PC-MRI Blood-Flow measurements

Visualization and Data Assimilation

de Hoon, N.H.L.C.

DOI
10.4233/uuid:5cac3120-16ac-4b06-977e-c703d37bb342

Publication date
2020

Document Version
Final published version

Citation (APA)
de Hoon, N. H. L. C. (2020). *PC-MRI Blood-Flow measurements: Visualization and Data Assimilation*. [Dissertation (TU Delft), Delft University of Technology]. https://doi.org/10.4233/uuid:5cac3120-16ac-4b06-977e-c703d37bb342

PC-MRI BLOOD-FLOW MEASUREMENTS

Visualization and Data Assimilation

N.H.L.C. de Hoon

PC-MRI Blood-Flow measurements

VISUALIZATION AND DATA ASSIMILATION

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus Prof. dr. ir. T.H.J.J. van der Hagen,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op maandag 14 september 2020 om 12:30 uur

door

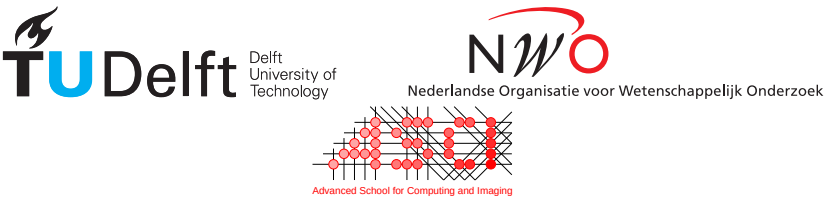
Niels Hendrikus Louis Cornelis DE HOON

Ingenieur in Computer Science and Engineering, Technische Universiteit Eindhoven, Nederland
geboren te Etten-Leur, Nederland.

Dit proefschrift is goedgekeurd door de promotoren.

Samenstelling promotiecommissie:

Rector Magnificus,	voorzitter
Prof. dr. A. Vilanova,	Technische Universiteit Delft / Technische Universiteit Eindhoven
Prof. dr. E. Eisemann,	Technische Universiteit Delft
Dr. A.C. Jalba,	Technische Universiteit Eindhoven
Onafhankelijke leden:	
Prof. dr. Ing. A. Hennemuth,	Charité - Universitätsmedizin Berlin, Germany
Prof. dr. ir. A.J. Nederveen,	Amsterdam UMC
Prof. dr. I. Hotz,	Linköping University, Sweden
Prof. dr. ir. M. Breeuwer,	Technische Universiteit Eindhoven
Dr. F.M. Vos,	Technische Universiteit Delft
Prof. dr. ir. B.P.F. Lelieveldt,	Technische Universiteit Delft / Leids Universitair Medisch Centrum, reservelid



This work was carried out in the ASCI graduate school.
ASCI dissertation series number 413.

Printed by: TODO

Front & Back: by N.H.L.C. de Hoon

ISBN TODO 000-00-0000-000-0

An electronic version of this dissertation is available at
http://repository.tudelft.nl/.

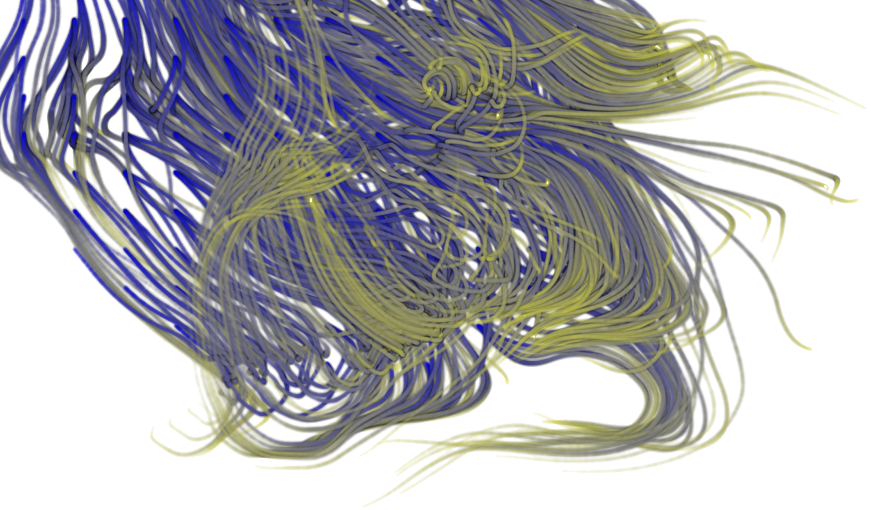
Contents

SUMMARY

Heart and vessel diseases, or cardiovascular diseases (CVDs), are globally the main cause of mortality and morbidity. The blood flow plays an important role in their occurrence and progression. Therefore, knowledge of the blood flow is of key importance to reduce and threat these diseases. This knowledge requires both high-quality data and an insightful visual representation. Using advanced imaging techniques, for example phase-contrast magnetic resonance imaging (PC-MRI), the blood flow in a patient can be measured. This technique provides patient-specific 3D data over time, that is, for every measured position, a so-called voxel, the velocity of the blood is measured in all three directions. From these three values per voxel, a vector can then be reconstructed that tells us how fast and in which direction the blood was flowing at the measured location. By doing this for multiple moments one can obtain this data throughout a heart cycle. Since this data is three dimensional and changing over time generating a visual representation of this data is challenging for multiple reasons. One such reason is occlusion where part of the visualization is hidden by the rest of the visualization. Another is visual clutter where the visualization is "too busy" and therefore unclear. Moreover, the measured data is subject to noise and artefacts which further hinder the visualization. In this dissertation, novel visualization approaches are presented that address these and other visualization challenges of PC-MRI data. Besides PC-MRI data, blood flow data can be created using computer simulation models, for example using computational fluid dynamics (CFD) models, that are based on physical models. For this usually the shape of the blood vessel is measured using imaging techniques which is in turn used to simulate the blood flow inside the vessel. However, both measuring and modelling the blood flow have their own advantages and disadvantages. For example, PC-MRI measurements suffer from the inevitable effects of measurement noise which causes the data to deviate from the actual blood flow in the patient. Simulations, on the other hand, require detailed input information and are based on model assumptions, and hence result in data that is not always representative for the patient, however, the resulting data does correspond to the physical model. In addition to visualization approaches, this work also presents novel methods that combine PC-MRI measurements and simulations such that the resulting data is both physically-plausible and patient-specific. The source code and executables related to this dissertation are made public and open source at https://gitlab.com/NielsDeHoon/QFlowExplorer.

SAMENVATTING

Hart- en vaatziekten (in het Engels afgekort tot CVD's) zijn wereldwijd de belangrijkste oorzaak van sterfte en morbiditeit. De bloedstroom speelt een belangrijke rol in het ontstaan en de voortgang van deze ziektes. Kennis van de bloedstroom is daarom van cruciaal belang om deze ziektes te behandelen en het aantal gevallen deze ziekten terug te dringen. Om deze kennis op te bouwen zijn zowel hoogwaardige gegevens als een inzichtelijke visuele weergave vereist. Met behulp van geavanceerde beeldvormingstechnieken, bijvoorbeeld fasecontrast magnetische resonantie beeldvorming (in het Engels afgekort tot PC-MRI), kan de bloedstroom bij een patiënt worden gemeten. Deze techniek genereert patiëntspecifieke 3D-gegevens tijdens een hartslag. Hiervoor wordt voor elke gemeten positie, een zogenaamde voxel, de snelheid van het bloed in alle drie richtingen wordt gemeten. Uit deze drie waarden kan per voxel vervolgens een vector worden gereconstrueerd die de snelheid en richting van het bloed op de gemeten locatie beschrijft. Door deze metingen meerdere malen te herhalen, kan men deze gegevens voor een volledige hartcyclus meten. Omdat deze gegevens driedimensionaal zijn en in de loop van de tijd veranderen, is het genereren van een visuele weergave van deze gegevens een uitdaging. Een van de redenen is occlusie waarbij een deel van de visualisatie wordt verborgen door de rest van de visualisatie. Een andere is visuele overprikkeling waardoor de visualisatie "te druk" is en daarom onduidelijk. Bovendien zijn de gemeten gegevens onderhevig aan ruis en artefacten die een duidelijke visualisatie verder belemmeren. In dit proefschrift worden nieuwe visualisatiebenaderingen gepresenteerd die deze en andere visualisatie-uitdagingen van PC-MRI-gegevens aanpakken. Naast PC-MRI-gegevens kunnen bloedstroomgegevens worden gemaakt met behulp van computersimulatiemodellen, bijvoorbeeld met zogehete computational fluid dynamics (CFD)-modellen, die zijn gebaseerd op fysieke modellen. Hiervoor wordt meestal de anatomie van het bloedvat gemeten met beeldvormingstechnieken en vervolgens gebruikt om de bloedstroom in het vat te simuleren. Zowel het meten als het modelleren van de bloedstroom hebben echter hun eigen voor- en nadelen. PC-MRI-metingen hebben bijvoorbeeld last van de onvermijdelijke effecten van metingsruis die ervoor zorgt dat de gegevens afwijken van de werkelijke stroming van het bloed in de patiënt. Simulaties vereisen daarentegen gedetailleerde informatie en zijn bebaseerd op modelaannames en resulteren daarom in gegevens die niet altijd representatief zijn voor de patiënt, maar de resulterende gegevens komen wel overeen met het fysieke model. Naast visualisatiebenaderingen presenteert dit werk daarom ook nieuwe methoden die PC-MRI-metingen als simulaties combineren, zodat de resulterende data zowel fysiek plausibel als patiëntspecifiek zijn. De broncode en uitvoerbare bestanden behorende bij dit proefschrift zijn openbaar en open source en te vinden op https://gitlab.com/NielsDeHoon/QFlowExplorer.



1

Introduction

Probably very early in the development our species, we realized the importance of blood for our survival. Simply through observations, it is clear that there is a link between blood loss and death. Early humans knew nothing of its true function but it sparked an ancient curiosity for blood and its "supernatural" powers, and became a symbol both of life and of death. As such, it is logical that blood plays a notable role in a lot of religious traditions. Unsurprisingly, it also played and still plays an important role in medicine.

According to the Ancient Greeks, the human body worked based on the amount of four liquids within the body: blood, phlegm, black bile and yellow bile, which they referred to as the four humours. According to this theory, also called humorism, a person was considered in good health if the four humours were in balance, an imbalance could result in disease. Hippocrates, although he was convinced that the theory of humorism was correct, is considered to be the first person to believe that diseases were caused by a natural cause and not by the supernatural. Later, Claudius Galenus, extended the theory of humorism. He was the first to distinguish the differences between dark oxygen-poor and bright oxygen-rich blood, however, he believed the difference in color to be due to the amount of "spirit" in the blood. However, since there is no visible connection between the blood vessels that transport oxygen-poor and oxygen-rich blood, he believed that both types of blood circulated in separate systems. Later, Ibn al-Nafis stated around 1242 that blood moves from the heart to the lungs, where it is mixed with air, and then back to the heart. Only in 1628 William Harvey published his (correct) theory on a singular circular system powered by the heart.

Traditionally, the main focus of research has been on anatomy, i.e. the structure and shape of the cardiovascular system. However, the *hemodynamics*, from the Greek *hemo* (blood) and *dynamic* (forces or movement), can influence the anatomy and vice-versa. As such, the blood flow also plays an important role in the occurrence and progression of *Cardiovascular Diseases* (CVDs) [1–3]. Therefore, it is important to have an understanding of the blood flow as well as the anatomy. For

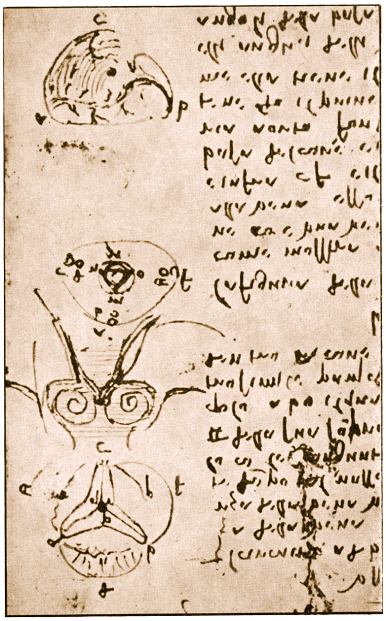


Figure 1.1: Schemetic sketches by Leonardi Da Vinci of the aortic valve and the blood flow.

Figure 1.2: An artist impression of the first published blood pressure measurement by Stephen Hales.

example, Leonardo da Vinci studied the aortic valve by reconstructing a glass model of a bull's aorta to see what the blood does in the heart when it pumps [4]. By introducing a flow into this glass model he was able to observe an estimate of the blood flow and theorize about the underlying principles [5]. His drawings of the observed flow, shown in Figure 1.1, may be the first visualization of the blood flow.

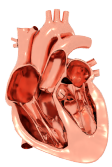
However, actual measurements of the blood flow within a living human being are much more difficult to obtain. The earliest quantitative (indirect) measurements of the blood flow were those of blood pressure. For example, the Ancient Egyptian physicians had already found a connection between the heart, pulse, and some diseases. Counting the pulse of a patient is still common practice in modern medicine and gives an indication of the blood pressure. However, measuring the actual pressure of the blood only began in the middle of the eighteenth century. In 1733 Stephen Hales was the first to publish a measurement of blood pressure. Using a tall vertical glass tube connected to a blood vessel of a horse the blood rises in the tube to a certain height and from there the blood periodically rises and falls a few centimetre with each of the horse's heart beats, as shown in Figure 1.2. Later, more sophisticate and non-invasive measuring devices were developed such as the sphygmomanometer, which is still the most commonly used device for measuring the blood pressure.

Nowadays, our knowledge of the cardiovascular system is much more complete and we can measure for example the anatomy of patients in high detail using imaging techniques such as *Computed Tomography* (CT) or *Magnetic Resonance*

Imaging (MRI). Yet, even today, heart and vessel diseases, or CVDs, are globally the main cause of dead [6–8]. Most of the deaths that are caused by CVDs are related to an unhealthy lifestyle of the patient, for example the diet of the patient and whether the patient smokes. Other CVDs are congenital, meaning that a person is born with these defects while others can be caused by trauma. Many CVDs are currently diagnosed based on the anatomy using medical imaging techniques. Measuring the blood pressure of a patient provides a physician with important, yet indirect information of the blood flow that can be used for the diagnoses of some CVDs. However, other, more modern measurement approaches, allow for a much more complete overview of the blood flow in a patient. Multiple options exist, for example, by using probes in the blood vessels or by using non-invasive imaging techniques. Of all the non-invasive methods, *Phase-Contrast Magnetic Resonance Imaging* (PC-MRI) provides the highest detail and most complete information, therefore, this thesis focusses on this data. The approach of PC-MRI for measure blood flow was first proposed by Paul R. Moran [9] in 1982 and the first full 3D velocity sensitive PC-MRI was published in 1996 by Wigström et al. [10], and since has been shown to provide valuable knowledge with respect to the occurrence and progression of some CVDs [1–3]. That is, PC-MRI can be used to measure three dimensional velocity data for multiple moments in time. Such collections of vector volumes are challenging to analyse for various reasons, for example, due to the amount of data. Since 1996 the PC-MRI technique has been improved, but still is rather novel. Therefore, doctors need to analyse what can and cannot be concluded from the data. New measurements require new analyses techniques to enable the discovery of new insides. Here both data processing techniques and visualization play a key role. This thesis provides novel techniques that help the experts with their analyses of PC-MRI data.

One of the challenges that experts face when working with PC-MRI data is the visualization. That is, the generation of images that present the users with the important features of the data they are (potentially) interested in. If we use a naive visualization by directly showing all the measured vectors, a lot of clutter and occlusion would occur and would not show any useful information, as demonstrated by Figure 1.3. Therefore, adequate visualization of flow data is essential for its understanding. A large body of research exists for the visualization of flow data, producing more useful visualizations, for example as shown by Figure 1.4 where healthy helical flow is shown in an aorta. Blood flow also has a highly relevant temporal aspect. The time-varying aspects are essential for the analysis, however, they are not visualized by the current state-of-the-art visualization frameworks for PC-MRI data. Furthermore, as for every measurement, PC-MRI is prone to various inaccuracies, such as artefacts. The inaccuracies of PC-MRI data are most often ignored in the visualizations used by the experts. However, this information could potentially change their analysis and conclusions. In this thesis we address this issue by providing various visualization methods for PC-MRI data that allow not only for common state-of-the-art visualizations, but, additionally, the visualization and exploration of the time-varying aspects and uncertainty of PC-MRI data sets.

Visualizing the blood flow in and near the heart generates extra challenges. The



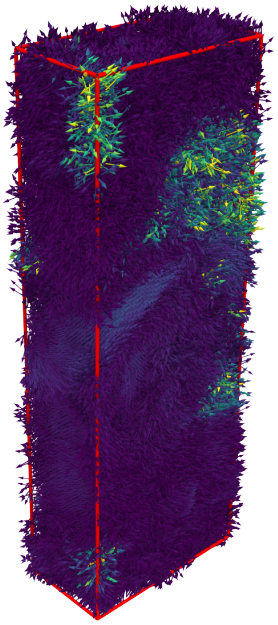


Figure 1.3: A naive visualization of the flow data.

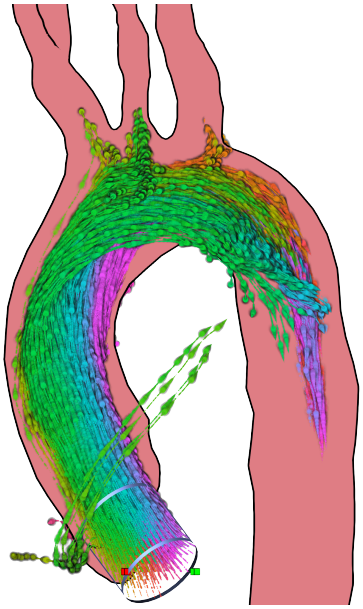


Figure 1.4: A visualization of the flow data geared to PC-MRI data. Compared to Figure 1.3 of the same PC-MRI data set, the details in the flow are shown much clearer.

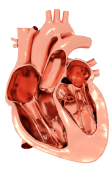
anatomy is less clear in the MRI data making it non-trivial to find regions of interest. Therefore, we introduce specific probes for the visualization of the blood flow in and near the heart using the measured PC-MRI data directly. To this end, we provide visualizations to visualize the anatomy of the heart as well as a way to select regions of interest within the heart.

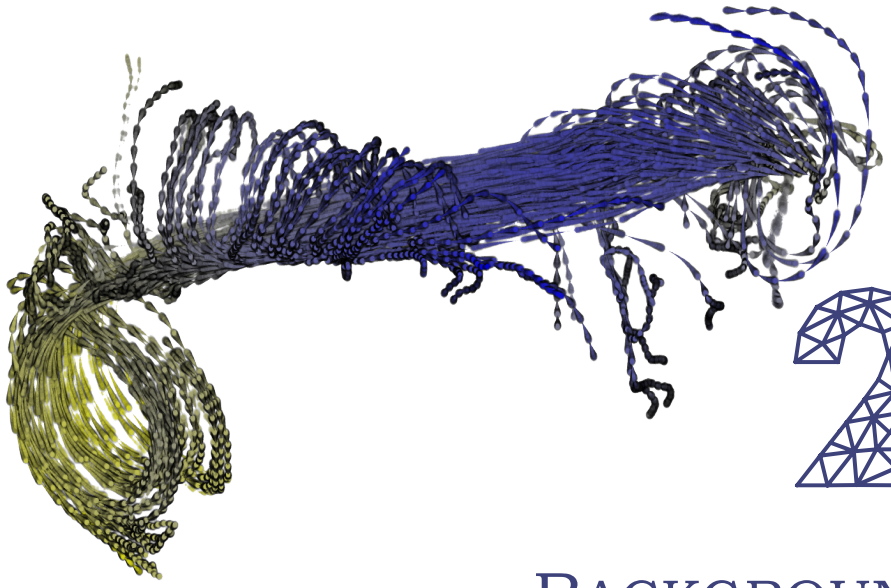
While PC-MRI data has a relatively low resolution compared to other non-invasive imaging methods, it provides more complete blood-flow information. Especially the number of measurements over time is low. Furthermore, as mentioned before, the measured data is prone to noise and artefacts which make the data deviate from the actual flow, resulting in flow data that is not physically plausible. Instead of measurements, physical models can be used, i.e. *Computational Fluid Dynamics* (CFD), to compute the blood flow within a given model anatomy. While such models do not suffer from artefacts and noise, there are modelling assumptions and simplifications. As such, the resulting flow information often deviates from the real patient specific flow. Using *data assimilation* the measured data can be combined with a physical CFD model and use the best aspects of both. Data assimilation is the process of combining observed (i.e., measured) data of a system with the corresponding scientific information (i.e., typically a mathematical or physical model of the system) to obtain a better estimate of the actual system. It attempts to use all available information to create the best match. We propose to use data assimilation for denoising, interpolation and extrapolation of measured PC-MRI data. That is,

we propose an approach that aims to increase the temporal resolution based on a physical model of the blood flow. It allows for the underlying physical knowledge to be taken into account for the interpolation. The information present in both measured data and physical model is used to deduce a more accurate and patient-specific representation of the blood flow. It also provides a new approach to denoise the data as well as improve the resolution both spatially and over time. Therefore, to the best of our knowledge we present the first approach that can provide experts with flow information that is both patient-specific, plausible and, additionally, has a high resolution.

The remainder of the thesis is structured as follows. To aid the reader in the understanding of the thesis some background knowledge is provided in Chapter 2. A brief introduction to the relevant cardiovascular anatomy is given and some related CVDs are explained. Furthermore, the acquisition of the data is explained as well as an introduction to fluid physics and its simulation. Since the work presented in this thesis focuses on a single topic, a global overview of previous work and the current state-of-the-art is given in Chapter 3. This provides the reader with a context to the methods presented in this thesis. The chapters that follow discuss the novel contributions within this thesis. The first two of these Chapters 4 and 5 focus on the visualization of PC-MRI data, whereas Chapter 4 focusses on a general approach to visualize and analyse both the effect of uncertainty and time-varying flow features, Chapter 5 deals specifically with the visualization of the blood flow in and near the heart. The following two chapters focus on data assimilation. Chapter 6 has a focus on denoising and improving the temporal resolution of PC-MRI data by using both the measured data and a physical model of the blood flow. This is extended on by Chapter 7 in which data assimilation is used to minimizing the difference between the physical model and the measured data, resulting in a physically-plausible description of the flow that is close to the measured data. Finally, in Chapter 8, the thesis is concluded by explaining the scientific and technical implications for society of the research presented in the thesis.

The 4D MRI blood-flow data sets used in this work were kindly provided courtesy of the division of Imaging Sciences, King's College London at St Thomas' hospital and Amsterdam Universitair Medische Centra (UMC).





2 Background

This chapter provides the reader with background information for understanding the remainder of this dissertation. This includes an introduction to the cardiovascular system and some of the corresponding diseases, an introduction to *Phase-Contrast Magnetic Resonance Imaging* (PC-MRI) and to *Computational Fluid Dynamics* (CFD).

2.1. The cardiovascular system

This section provides a brief introduction to the medical background for understanding the data used in this dissertation. An overview of the anatomical features can be found in Figure 2.1. Note that, as anatomic variations can occur, the most common anatomic structure is used as a basis in this Section.

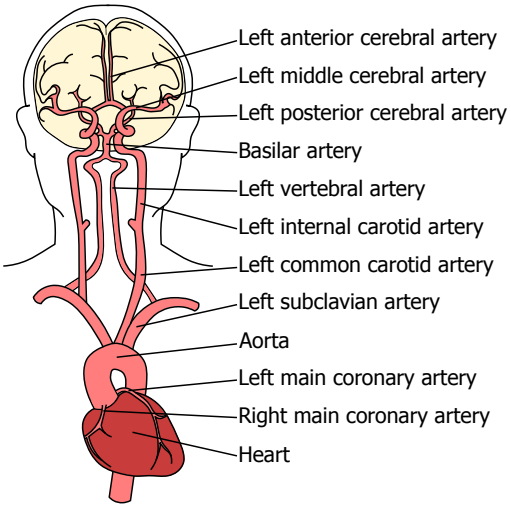


Figure 2.1: An overview of the anatomy of the section of the cardiovascular system this dissertation focuses on.

The cardiovascular system has three main components: the heart, the blood vessel and the blood itself and is responsible for the circulation of blood to the various parts of the body. The cardiovascular system consists of the *systemic circulation*, which provides the body with oxygenated blood and the *pulmonary circulation* through the lungs to oxygenates the blood. Moreover, the blood transports nutrients and waste products, e.g., carbon dioxide through the body. The blood vessels

are separated into arteries that come from the heart and veins that go to the heart. Arteries have a high oxygen level, and veins a low oxygen levels, with the exception of the pulmonary artery and pulmonary veins that are part of the pulmonary circulation. The pulmonary artery leads oxygen-poor blood to the lungs, whereas the pulmonary veins lead oxygen-rich blood from the lungs back into the heart. The aorta is an artery that leads oxygen-rich blood to the body, while the inferior and superior vena cava lead oxygen-poor blood back into the heart. The time period in which the heart pumps blood is called *systole*, which is followed by the *diastole* during which the heart refills with blood. As a result the blood flow speed is highest in the aorta during *peak systole* namely between 200 and 250 cm/s. An efficient blood flow is essential, since the main task of the heart is to provide oxygenated blood to the rest of the body.

In this dissertation the focus is on the heart, aorta and so-called intracranial blood flow.

2.1.1. The heart

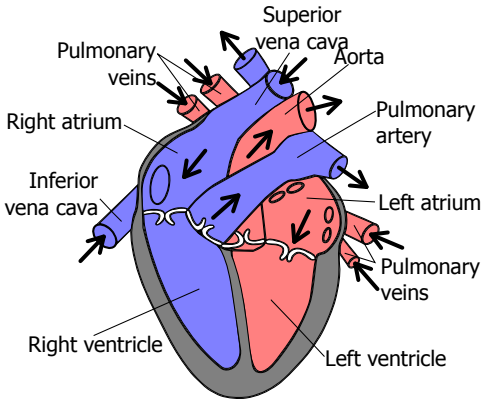


Figure 2.2: Anatomy of the human heart. Red indicates oxygenated blood, while blue indicates oxygen-poor blood. The arrows show the flow direction.

The heart is about the size of a fist and is the pump of the cardiovascular system. It pumps deoxygenated blood into the lungs to oxygenate the blood and then pumps it to the rest of the body. Figure 2.2 shows the anatomy of the human heart and the general blood-flow directions. The heart is separated into a left and a right side by a muscle called the septum and has four chambers. The two top chambers are the two so-called atria, while the two chambers at the bottom are the ventricles. The left and right atria receive blood from different sources, respectively the lungs and the rest of the body. When the atria contract, the blood inside is pushed through the valves and fills up the ventricles. The right and left ventricle are, respectively, responsible for pumping the blood out to the lungs and the rest of the body. Here, the tricuspid and mitral valves serve as one-way doors: blood should not flow back in when the right and left ventricle respectively when they contract. Then, when the ventricles contract, blood is pushed through the pulmonary valve into the lungs and through the aortic valve into the rest of the body. These valves serve as one-way doors as well. However, in case of a so-called mitral valve regurgitation, the blood can flow back from the left ventricle into the left atrium. This leakage can increase blood volume and pressure in the left atrium and the pulmonary veins. In severe cases, the heart can be enlarged to provide flow of blood in the aorta, potentially causing heart failure.

Furthermore, the increased pressure in the left atrium and the pulmonary veins can result in congestion (or fluid build-up) in the lungs.

2.1.2. The aorta

Every heart cycle, the heart pumps blood into the *aorta*, the largest artery in the human body which transports this oxygen rich blood from the heart to the other organs. Figure 2.3 gives an overview of the anatomy of the upper part of the aorta.

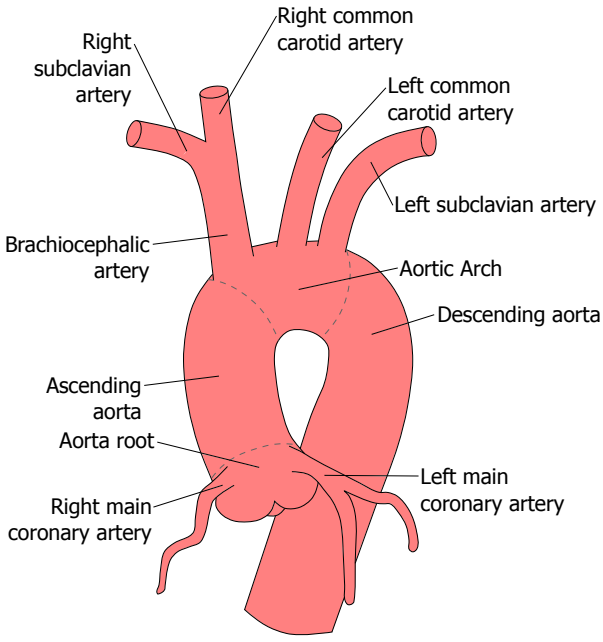


Figure 2.3: Anatomy of the upper part of the human aorta. The different sections of the aorta are separated by dotted lines.

The aorta is connected to the heart through the *aorta root*. It mainly consists of the aortic valve that regulates the blood flow into the aorta. The valve consists of three valve leaflets that open and close during a heart cycle, these leaflets also prevent regurgitations, i.e., blood flowing back from the aorta into the heart. Each of the leaflets has a corresponding sinus or cusp in which vortices form after peak systole that allows for smooth opening and closure of aortic valve [5, 11]. These vortices also optimize the flow and minimize the stress on aortic valve leaflets [12–15]. The aorta root has two coronary arteries that provide blood to the heart.

After the aorta root, the *ascending aorta* begins and ascends upwards until the *aortic arch*. This aortic arch is where the aorta bends downwards, the flow typically has a right handed helical flow pattern in healthy individuals [11]. Three major vessels originate from aortic arch, namely the brachiocephalic artery (or innominate artery), the left common carotid artery and left subclavian artery. The brachiocephalic artery splits into the right subclavian artery and the right common carotid artery. The left and right carotid arteries provide the corresponding sides of the brain with blood, while the subclavian arteries provide the arms with blood.

The aortic arch is followed by the *descending aorta* which continues downwards and provides blood the remainder of the body. This descending aorta has two parts: the *thoracic aorta* and *abdominal aorta*, which respectively are in the thorax (chest) and abdomen (the part of the body between the thorax and pelvis).

2.1.3. Intracranial vessels

The blood vessels within the skull, or the *intracranial blood vessels*, are involved in distributing the blood over the brain. From the aorta blood is supplied to the brain via 4 arteries, namely, the two vertebral arteries and two common carotid arteries. The two vertebral arteries branch from the brachiocephalic and subclavian arteries and come together in the basilar artery.

Circle of Willis

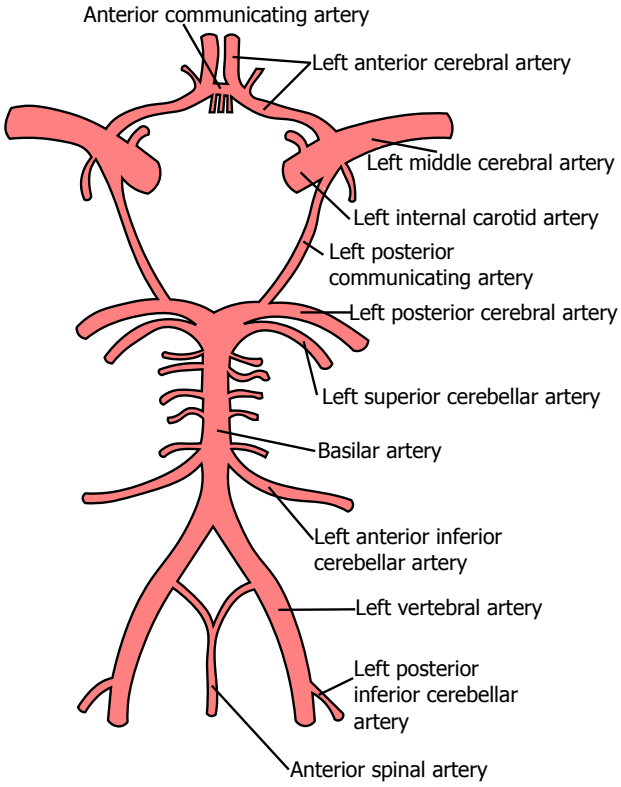
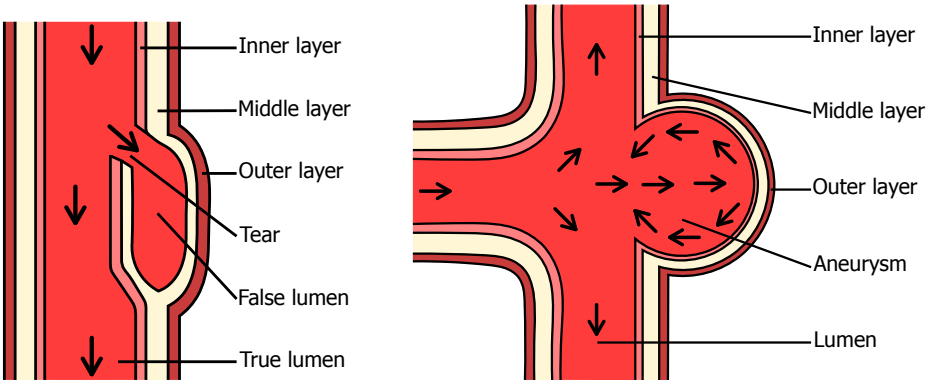


Figure 2.4: The circle of Willis and some of the connected vessels.

The circle of Willis, or cerebral arterial circle, is a relatively small circular vessel structure that supplies blood to the brain and the surrounding structures. The following arteries are considered to be part of the Circle of Willis: the anterior cerebral arteries, the anterior communicating artery, the internal carotid arteries, the posterior cerebral arteries and the posterior communicating arteries. Note that middle cerebral arteries are not considered part of the circle. An overview of the circle of Willis and the connected vessels is shown by Figure 2.4. Since the blood can take multiple routes to reach a destination, a circular structure seems to be redundant. However, in case of a blocked or narrowed artery, the remaining arteries can provide the brain with enough blood. The arteries of the circle of Willis typically have small diameters, i.e. smaller than 1 centimeter.

2.1.4. Cardiovascular diseases

Many diseases can affect the cardiovascular system. In this Section, two diseases important for understanding the remainder of the dissertation will be introduced and discussed.



(a) A schematic overview of a dissection in a vessel. The arrows indicate a possible blood flow in this scenario.

(b) A schematic overview of a aneurysm in a vessel. The arrows indicate a possible blood flow in this scenario.

Figure 2.5: A representation of two cardiovascular diseases that can affect both the anatomy and blood flow.

Dissection

The vessel wall consists of layers, which under certain conditions can split. Such a split, or *dissection*, occurs when a tear in a vessel wall separates the layers of the wall and allows blood to flow between the layers of the wall. This creates two direction for the blood to flow in, into original direction through the *true lumen*, or between the layers of the wall into the *false lumen* as shown in Figure 2.5a. A dissection can cause abnormal blood flow and deform the vessel, potentially disrupting the blood flow to organs if the size false lumen prevents enough blood to flow through the true lumen.

Dissections can occur in various blood vessels. For example in the aorta. While such an *aortic dissection* is relatively rare, the consequences are often severe [1].

Aneurysm

When a weak spot on the vessel wall causes an outward expansion of the vessel, a so-called aneurysm occurs, see Figure 2.5b for a schematic example. With an aneurysm, the probability of a rupture of the vessel wall increases, and can lead to an internal bleeding. Depending on the size of the region of the vessel wall involved in the aneurysm, they can have various shapes and sizes. Any blood vessel can be prone to developing an aneurysm, however, when they for example, occur in the circle of Willis or the aorta they can have severe consequences.

2.2. Measuring blood flow

The blood flow of a patient can be measured in a non-invasive manner through various imaging techniques. *Computed Tomography* (CT) and (rotational) X-ray are imaging techniques that are often used, for example for anatomical imaging.

Both can be used to derive blood-flow information using contrast agents. However, the patient is exposed to radiation during the scan and the derivation of the flow data is rather limited.

Ideally we would want to obtain the whole *velocity field* of the blood flow, i.e. we want to know the velocity of the blood for every position within the patient. Note that in physics the term *field* represents an assignment of a quantity to every position in a selected space. This quantity can for example be a scalar or a vector. For example the density of a fluid is a scalar, while the velocity of a fluid is a vector.

To measure the blood flow Doppler ultrasound can be used and is currently the clinical standard. It allows for high spatial and temporal resolutions and is cost effective [16]. However, the field-of-view is relatively small, meaning that only a small region within the patient can be shown at a given moment. Moreover, the method is relatively prone to noise and it does not provide a dense 3D vector field.

Magnetic Resonance Imaging (MRI), and more specifically PC-MRI, is capable of measuring a structured 3D vector field over time. Of all available non-invasive techniques it provides the highest detail and most complete information. As such, PC-MRI data is used in this thesis.

2.2.1. PC-MRI

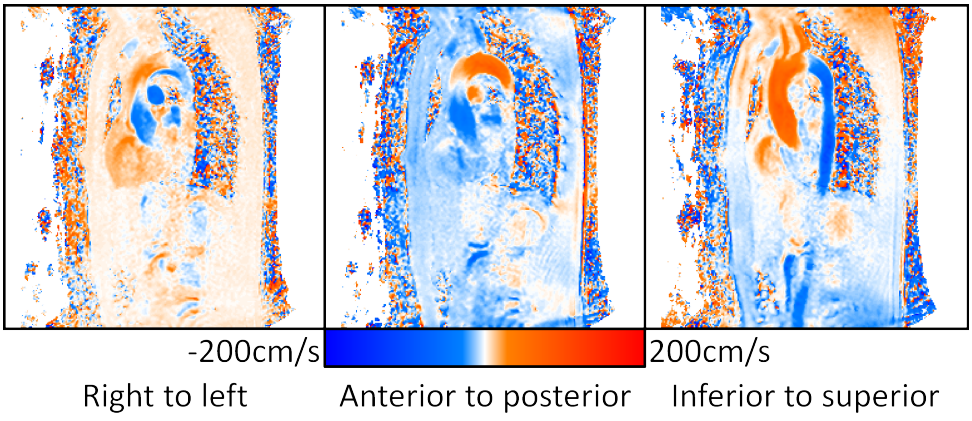


Figure 2.6: Three PC-MRI slices at peak systole showing the three measured velocity components. Blue and red respectively represent negative and positive values, while white indicates values near zero.

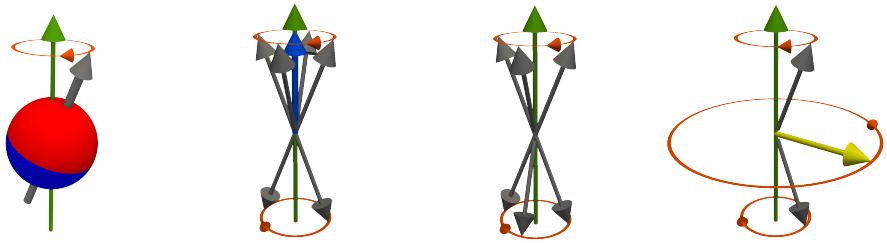
MRI allows imaging of both the anatomy as well as the blood flow. The spatial and temporal resolution is typically lower than that of Doppler ultrasound and the costs are higher, however, the signal-to-noise ratio and field-of-view are better. While PC-MRI is not commonly used in a clinical setting, research indicates that it provides insights in the blood flow and helps to determine the importance of various flow features [2, 5, 11, 17–19]. Typically the spatial resolution is in the order of 1.0 to 3.0mm. A full heart beat is commonly measured using 20 to 25 measurements over time, these so-called *phases* result in a temporal resolution of 20 to 50ms. Especially the temporal resolution is coarse since healthy blood

flow can reach speeds of 200cm/s and higher, meaning that the blood can move 10cm between two measurements. Higher resolutions can be acquired at the cost of higher noise levels or by longer (impractical) scanning time [2]. The maximum speed that can be measured is given by a user-defined acquisition parameter: the *velocity encoding speed* (v_{enc}) and typically ranges from 100 to 200cm/s. Figure 2.6 shows the three velocity components of a data set at peak systole. The images in this paper are generated from PC-MRI data sets of healthy volunteers and patients.

2.2.2. Acquisition of MRI data

MRI exploits the magnetic properties of protons, neutrons, whole nuclei and electrons and their so-called spin. For imaging, the hydrogen nucleus, which has a single proton, is often used because its high occurrence in various tissues, e.g., water and fat. The particles, e.g. hydrogen protons, have a magnetic moment, which makes it act like a tiny magnets with a north and south pole on a macroscopic scale. Furthermore, the particle has an intrinsic angular momentum that is inherent to any fundamental particle: the so-called spin. Note that the particle is not actually spinning. The spin can be represented by a vector around which the particle spins and has a magnitude and an orientation, similar to the north pole of a magnet. Normally, every particle has a random orientation, however, when a strong external magnetic field is applied, for example by an MRI scanner, the particles align their orientation parallel or anti-parallel with the magnetic field and the particle starts precessing. Precession here means that the orientation of the particle is rotating around the direction of the magnetic field at a fixed angle, as shown by Figure 2.7a. This precession happens at a specific frequency depending on the strength of the magnetic field. Most particles will have a parallel alignment with the magnetic field causing a longitudinal magnetization in the direction of the magnetic field, as shown by Figure 2.7b, which cannot be measured.

When radio waves with a frequency equal to the precession rate are emitted the orientation of some of the particles with a parallel alignment will become anti-parallel. Therefore, if the right amount of energy is used, the longitudinal magnetization can be cancelled out, as depicted by Figure 2.7c. Moreover, the radio waves will synchronize the spin of the particles, that is the phase of the precession will be equal for all particles, i.e., the particles will start to *resonate*. This causes the longitudinal magnetization to precess in an orthogonal direction to the magnetic field made by the scanner, this is the so-called transverse magnetization, as shown by Figure 2.7d. This transverse magnetization can be measured. When the scanner stops emitting the radio waves, the particles will fall back and the net transverse magnetization will become zero. This is the so-called T2 relaxation and no energy is emitted. When the so-called T1 relaxation occurs, some particles move from the anti-parallel orientation to back the parallel orientation and the longitudinal magnetization will restore which causes a transfer of heat energy, returning to the state shown in Figure 2.7b. More precisely, T1 is defined as the time when 63% of the longitudinal magnetization has recovered and T2 is is defined as the time when 63% of the transverse magnetization has decayed. Since different tissues have different characteristics the difference between T1 and T2 relaxation can be used



(a) The orientation of the spin (black arrow) of the particle is rotating around the direction of the magnetic field (green arrow) at a fixed angle.

(b) The precession of multiple particles. The net longitudinal magnetization (blue arrow) is in the direction of the magnetic field (green arrow).

(c) Under the influence of radio waves with the right frequency and energy, the net longitudinal magnetization is zero.

(d) The same radio waves make the spins synchronize and lead to the net transverse magnetization (yellow arrow) that can be measured.

Figure 2.7: Figure (a) shows the precession of a single particle. The particles are grouped together in Figures (b), (c) and (d) show schematically the steps of a single MRI sequence. The dark grey arrows indicate the spin of the particles while the orange disks indicate the precession trajectories.

to distinguish different tissue types. Other scan parameters are the time between the pulses of radio waves and measuring the return signal, respectively *Repetition Time* (T_R) and *Echo Time* (T_E).

If a homogeneous magnetic field would be used, all the particles within the field have the same precession rate and orientation. Therefore, the location should be encoded in the signal. By using a series of gradient electric coils the local strength of the magnetic field can be changed. Since the precession rate is determined by the magnetic field strength, a slice can be selected by using a gradient magnetic field, say in the z direction. This slice will consist of the only particles that have a precession rate that matches with the radio wave frequency, the other slices have a higher or lower precession rate due to the magnetic field. A row within this slice can then be selected by using another gradient magnetic field in another direction, say the y direction. To encode a location within the row, a another gradient magnetic field in x direction is applied. This last gradient field also changes the precession rate of all the locations within the row. Then the gradient magnetic field in x direction is turned off, causing all particles in the row to have the same precession rate again. However, the phase of the precession, i.e., the orientation, is shifted. Therefore, the specific locations within a row are encoded and can be deduced based on the phase and frequency of the return signal. By doing so for all rows and slices, a 3D volume can be constructed that consists of a regular grid which consists of so-called *voxels*. Each such voxel is cell of the grid and has a value. More details on the acquisition of MRI are provided by Berger [20].

2.2.3. Acquisition of PC-MRI data

The previous section explained the general concept of MRI. However, one assumption for spatial encoding is that the tissue does not move. This is not the case for the beating heart and the blood flowing through the vessels. To measure the velocity of tissue, PC-MRI not only uses a spatial encoding but also encodes the position in time. It applies the so-called spin-phase effect referring to chances in the precession angle (or phase) that occur when a particle moves through a gradient. If a particle moves while these gradients are applied, it will gain or lose phase compared to stationary tissue.

To derive the velocity, first, the main magnetic field is used to align all particles with the magnetic field, i.e., all particle stationary or moving have no phase change. Next a bipolar gradient pulse is used, i.e., two gradients with the same magnitude but opposite gradient direction are switched on and off in succession for an equivalent amount of time. If tissue is stationary, the bipolar gradient has no effect, but, if it moves it produces a phase shift of the transverse magnetization since the particles experience a difference between the two gradients when they move in the direction of the gradient. The amount of phase shift depends on the velocity of the particles along the direction of gradient. This shift in phase is linearly related to the velocity of the particles in a voxel. Therefore, by measuring the phase, the velocity in the direction of the bipolar gradient pulse can be measured.

Since phase shifts can be caused by, for example, field inhomogeneities, the measurement is repeated with an inverted bipolar gradient. The two data sets are subtracted and the phase difference that remains is used for a voxel-wise calculation of velocities. By measuring in x, y and z direction we obtain a vector volume. This is the so-called *Phase-Contrast Angiography Phase* (PCA-P) volume, which provides the type of quantitative velocity data shown in Figure 2.6. Both data sets also yield information about the signal magnitude for each voxel. These signal intensities are processed into an anatomic image: the magnitude image. When the two data sets are subtracted this yields a scalar field that is indirectly related to the velocity, the resulting volume is the so-called *Phase-Contrast Angiography Magnitude* (PCA-M) volume with a high *Signal-to-Noise Ratio* (SNR) within the vessels.

Phase shifts have a fixed range of $[-180°, 180°]$ degrees. Therefore, the peak velocity that can be measured is bounded and should be given by the user prior to the scan such that the expected peak velocity, v_{enc}, corresponds to a phase shift of $\pm 180°$. This v_{enc} is usually defined in cm/s and should be selected with care, for velocities larger than v_{enc} the phase can wrap around which causes artefacts. On the other hand, a v_{enc} that is too high will be less sensitive to small variations in velocity.

To have a model of the anatomy of the vessel, a segmentation is usually generated using marching cubes on the *Temporal Maximum Intensity Projection* (TMIP) which is less sensitive to temporally changing noise [21]. The resulting mesh is then manually smoothed using mesh-editing software.

More details on the acquisition of PC-MRI data are given by Lotz et al [22], Markl [2] and Gasteiger [23].

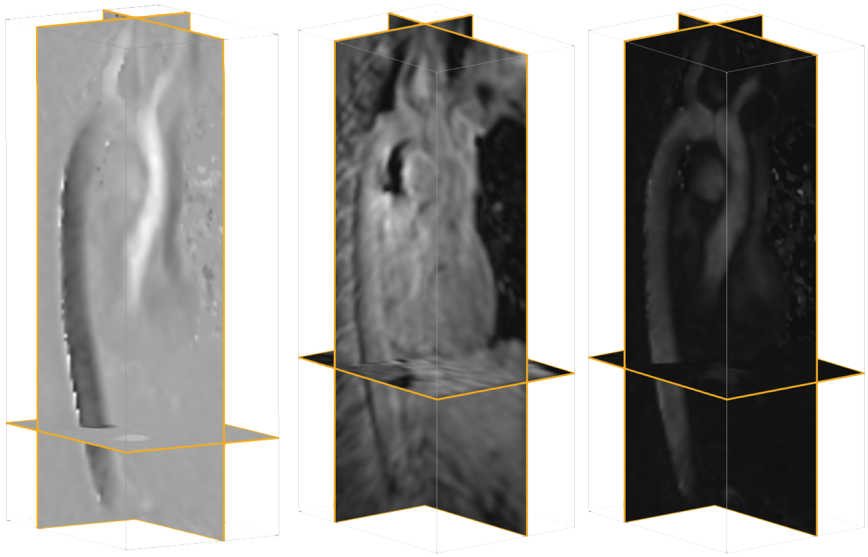


Figure 2.8: Respectively, one of the three velocity components (inferior to superior) (a), the magnitude (b) and the corresponding signal to noise ratio (c), computed using the method by Friman et al. [24, 25]. All at peak systole of a healthy volunteer.

Uncertainty in PC-MRI data

In this Section, an estimation of the PC-MRI measurement noise is provided. Note, that this dissertation does not focus on the derivation of the uncertainty models, as this is considered out of scope. However, an overview of an estimate of the uncertainty caused by measurement noise is given since it is used in this thesis. For such an estimate of the uncertainty caused by measurement noise, the method presented by Friman et al. [24, 25] provides a voxel-wise estimate of the uncertainty. Throughout this section vectors and matrices are represented in boldface.

Friman et al. [24, 25] estimate the SNR as $SNR = P_s/P_n$, with P_s and P_n denoting signal and noise power respectively. The Rician noise in MRI data can be approximated by Gaussian noise [26] for a high signal-to-noise ratio ($P_s/P_n > 5$). This is normally the case for voxels containing blood [24, 25]. To compute the SNR, we need to estimate the signal strength A and the noise variance σ^2. For every voxel i, the signal strength is given by

$$A_i = \frac{1}{4}\left(\left|I_i^O\right| + \left|I_i^x\right| + \left|I_i^y\right| + \left|I_i^z\right|\right). \tag{2.1}$$

Here I_i^k, $k \in \{O, x, y, z\}$, is the reconstructed image at voxel i, the superscript O represents the magnitude image, while the superscripts x, y and z denote the different velocity components. Note that the SNR is based on the anatomy and velocity. By doing so low velocity features of the flow within the vessel, such as vortices [5], can still have a relatively high SNR. From this, we derive a mask Ω that only contains

high signal strength voxels, i.e., the non-air voxels. For these voxels, the Rician distribution of the noise in the measurements can be approximated by a Gaussian distribution. The noise variance for all voxels is estimated by

$$\sigma^2 = \frac{1}{|\Omega| - 1} \sum_{i \in \Omega} \left(\frac{1}{3} \sum_{k \in \{0,x,y,z\}} \left(I_i^k - A_i \right)^2 \right), \tag{2.2}$$

where $|\Omega|$ denotes the number of voxels in Ω. Note that we assume that the noise variance is equal for all voxels in the volume, in contrast to the signal strength A, which is derived for each voxel, similar to Friman et al. [24]. Figure 2.8 shows the resulting SNR (A/σ^2) at peak systole for a healthy volunteer. For our data sets, we found an average SNR of approximately 10.

From the SNR, a covariance matrix $\mathbf{C}$ is constructed for each voxel i as follows

$$\mathbf{C}_i = \frac{{v_{enc}}^2}{\pi^2} \cdot \frac{\sigma^2}{A_i^2} \cdot \begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix}, \tag{2.3}$$

From the covariance matrix it follows that for each voxel i we have the following Gaussian distribution V_i per measured velocity component v_i^x:

$$V_i \sim \mathcal{N}\left(v_i^x, \frac{{v_{enc}}^2}{\pi^2} \cdot \frac{2}{A_i^2} \right). \tag{2.4}$$

This provides us with a per-voxel probability distribution that we can sample. Moreover, it gives us a value that represents the certainty of the measured data.

2.2.4. Phantom data

Similar to the glass reconstruction of a bulls aorta by Leonardo da Vinci a replica of a vessel can be reproduced based on a 3D scan of a vessel structure in a patient. Such anatomical models, so-called *phantoms*, are often made out of glass. Phantom data is often used to allow for controlled PC-MRI measurements that are necessary for validation or for small blood vessel structures that are too small to measure reliably in a patient such as aneurysms. Flow is created inside the phantom using specialized pumps and the inflow can be based on the velocity profile measured in the patient. One of the advantages of phantom data over patient data is that the flow can be measured at a higher resolution and with a higher SNR, since the imaging time can be increased. Therefore, phantom data has a relatively low amount of measurement noise. Furthermore, by using a model the exact location of the vessel wall is known and hence less (motion) artefacts can be expected.

2.3. Computational fluid dynamics (Navier-Stokes Equations)

While PC-MRI data is patient specific, i.e., the measured flow describes the actual flow in the patient, the measurements are also prone to noise and measurement

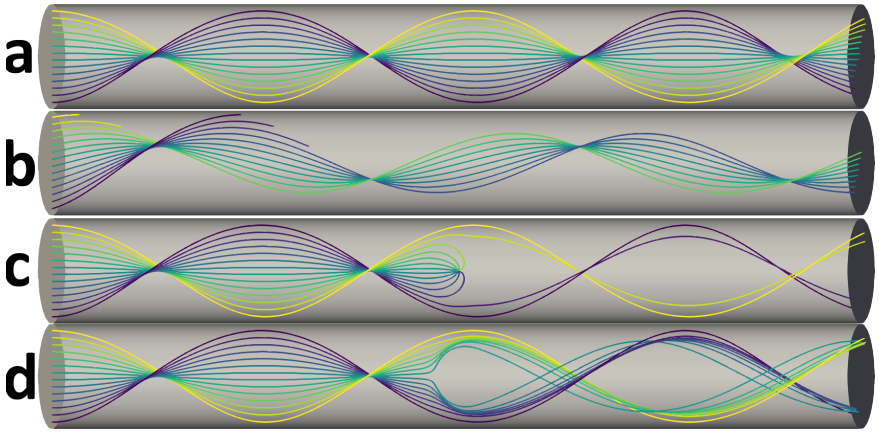


Figure 2.9: Inconsistencies in a measured flow data can have a big impact on flow visualization. Respectively, from top to bottom, (a) the original synthetic data for reference, (b) a constant offset in velocity, (c) a sink and (d) a source. Here, in image (b) the vector field has an offset in one direction causing the flow to leave the domain. Images (c) and (d) suffer from a single source of divergence. Note that the number of parallel integration lines decreases in all three scenarios compared to the baseline as shown by image (a).

artefacts as was described in Section 2.2.3. This has an impact on analyses and visualization. For example, the flow data could have divergence, meaning that the flow field has regions where the flow is moving away from a single point, i.e. a "source" or towards a single point, i.e. a "sink". Due to the way PC-MRI is measured, the flow data might have an offset towards a certain direction, e.g. causing flow to leave the segmentation. Each of these artefacts change the flow visualization substantially, and as such, the flow could be interpreted incorrectly, as shown by Figure 2.9. Moreover, for quantitative analysis, these artefacts must be taken into account with care. E.g., for quantitative particle tracing, where particles are emitted and counted to determine, for example, the percentage of blood volume that is ejected by the heart in one heart cycle and, thus, leaves the heart adequately. In this case, if particles are leaving the domain or get stuck in a sink, such percentages will be negatively effected and therefore less reliable. While the data is a measurement of a physical phenomena, the data is likely to have some degree of noise and artefacts. Therefore, to obtain physically correct data often a physical model is used to model the hemodynamics, i.e., the dynamics of the blood flow. For this a physical description of fluid is necessary. This can be done by describing how the velocity of the fluid changes over time. CFD is an extensive field of research that focusses on using physics to compute fluid behaviour. In physics, the **Navier-Stokes** equations describe fluid flows. These equations consider the fluid to be a continuous field or *continuum*, instead of a set of a very large number of discrete particles, e.g. blood cells. By doing so, it describes the interactions of the fluid flow on a higher level. We will use the following notation: vectors are denoted in bold and we use $\mathbf{u}$ to describe the velocity field.

For the density of the fluid, essentially the mass of the fluid per volume, we use

ρ.

In general there are 4 components that influence how a velocity field (or flow) of a fluid changes over time: *advection* or *convection*, *diffusion* and *internal and external forces*. Note that other influences can play a role, for example, heat distribution or heat advection, but this can be considered out of scope for the scenarios in this work.

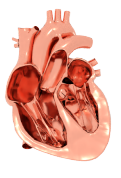
The velocity field changes itself through time by **advection**, that is, the velocity field transports itself over the velocity field. When we think of a lot of small particles, this would be the moving particles that "push" other particles: the velocity transports the velocity field through time. Mathematically, we can describe this movement of the velocity field as $\mathbf{u} \cdot \nabla\mathbf{u}$, where $\nabla\mathbf{u}$ denotes the gradient of $\mathbf{u}$. **Diffusion** occurs when we consider the friction between the molecules of the fluid, and is the reason for some fluids to be more **viscid**. The viscosity of a fluid determines how much the fluid resists deformation due to shear stress or tensile stress. Nearly all fluids have a non-zero viscosity, besides superfluids under special conditions. Fluids with a high viscosity, for example syrup move differently compared to fluids with a lower viscosity such as water. Since the viscosity of a fluid is a material property that can be measured, it is often represented by the dynamic viscosity μ or by the kinematic viscosity $\nu = \mu/\rho$. The term $-\mu\nabla^2\mathbf{u}/\rho$ accounts for changes of the velocity field over time due to the viscosity.

Since the conservation of mass is a fundamental physical law, the fluid must preserve its mass at any time. That is, no mass should disappear or appear. For this, we often consider the fluid to be **incompressible**, that is the volume of the fluid does not change. More precisely, the amount of fluid that enters a given volume is equal to amount of fluid that leaves the volume. For example, if the given volume is a cube, the amount of fluid entering and leaving is determined by the areas of the faces of the cube times the local velocity of the fluid. Mathematically, incompressibility means that **divergence** of the velocity field is restricted to be zero at every position: $\nabla \cdot \mathbf{u} = 0$. **External forces**, for example gravity $\mathbf{g}$, that act on the fluid which are usually considered to be constant. When such external and body forces are applied, fluid pressure builds up. However, even in this case the fluid must remain incompressible, so a correction of $\frac{1}{\rho}\nabla p$ is applied to enforce the incompressibility. Note that, pressure p represents the amount of force applied by the fluid onto itself, hence, differences in pressure cause the fluid to flow. If we combine all these terms that describe the change of the velocity field over time, we get the Navier-Stokes Equations for incompressible fluids:

$$\frac{\partial \mathbf{u}}{\partial t} = -\mathbf{u} \cdot \nabla\mathbf{u} + \mu\nabla^2\mathbf{u}/\rho - \frac{1}{\rho}\nabla p + \mathbf{g} \tag{2.5}$$

$$\nabla \cdot \mathbf{u} = 0, \tag{2.6}$$

Note that for simplicity, often blood is model as an inviscid fluid, and thus the viscosity term is neglected. Moreover, the effect of gravity on the blood flow is often



negligible, therefore in many cases we use the simplified form:

$$\frac{\partial \mathbf{u}}{\partial t} = -\mathbf{u} \cdot \nabla \mathbf{u} - \frac{1}{\rho} \nabla p \tag{2.7}$$

$$\nabla \cdot \mathbf{u} = 0, \tag{2.8}$$

2.4. Simulation of fluids

The previous section described the physical fluid laws using various vector and scalar fields. These fields represent continuums of physical quantities, e.g., velocity or pressure, for all points in both space and time. Naturally, a computer cannot store and update a continuum for all positions and for all moments in time, as these would require an infinite amount of values. This is where the field of CFD plays an important role as this field deals with numerical analyses and data structures for solving problems that involve fluid flows using the computer. CFD methods use a set of representative positions/regions and representative moments in time to model a fluid. That is, the continuous fields are *discretized*, i.e., these fields are transferred into a fixed number of discrete counterparts. Naturally, whenever a continuous field is discretized, an error is introduced since not all information in the field is taken into account. Different modelling approaches use different types of discretization and each comes with their own benefits and drawbacks. Common discretization approaches are categorized as in the following categories: mesh-based approaches and mesh-free approaches. Examples of mesh-based (Eulerian) approaches are the *Finite Difference Method* (FDM) [27] and *Finite Element Method* (FEM) [28], while examples of mesh-free (Lagrangian) approaches are the *Smoothed Particle Hydrodynamics* (SPH) [29] method and 'vortex particle method' [30]. For the simulation of blood flow various approaches are being used [31]. In this work, since the measured data uses a regular grid, we use simulations that use a grid similar to measured grid. Naturally, anatomy that does not align with the grid is non-trivial to deal with, however, a regular grid trivially maps to the measured data. Therefore, we focus on simulations that have an Eulerian basis.

2.4.1. Fluid simulation in computer graphics

The field of *computer graphics* focuses on the generation of pictures and films created using computers which are used for example in movies or games. This includes the generation of fluid effects, such as water and fire. For this, various computer graphics simulation methods exist that allow for the generation of realistic fluids that are based on the Navier-Stokes equations. Naturally, the accuracy of a simulation model is bounded by the available computation time. Within computer graphics, both Eulerian and Langrangian methods are commonly used.

Lagrangian techniques can exhibit more small-scale features that are not bound by a fixed grid cell size, since they model fluids using particles. Incompressibility is, however, hard to ensure.

One of the first stable Eulerian methods in computer graphics was introduced by Stam [27]. In this seminal work, Stam presented a stable semi-Lagrangian fluid sim-

ulation with regard to advection, enforcing incompressibility and mass conservation using an Eulerian grid representation. However, such purely Eulerian approaches lack details smaller than a grid cell.

To exploit the best of both the Eulerian and Lagrangian approaches, hybrid simulation techniques have been proposed, using both particles and grids. A commonly used hybrid approach is the *Fluid Implicit Particle* (FLIP) method, and extension of the *Particle In Cell* (PIC) method, introduced by Brackbill et al. [32] and popularized for use in computer graphics by Zhu et al. [33], Batty et al. [34, 35] and Bridson [36]. This hybrid approach preserves details and ensures incompressibility.

Fluid Implicit Particle (FLIP) method

In this thesis we use the FLIP approach for fluid simulation. As mentioned before, we model blood as an inviscid fluid. Although it is common to neglect viscosity when simulating water for computer graphics applications [36] and the simulation of blood [37], our choice of excluding viscosity is also based domain-specific characteristics; e.g., unreliable location of the vessel walls and the fact that viscosity effects are minimal at high-speed blood flow through the main vessels. Moreover, due to the discretization and averaging the simulation method has some inherent numerical viscosity. We use a standard operator-splitting approach that allows separating the different terms of the Navier-Stokes Equation given in Section 2.3 into multiple terms, calculated in sequence, independently of one another. Note that the current velocity field $\mathbf{u}^n$ is assumed to be known and we want to compute the velocity field after one time step, hence we want to obtain $\mathbf{u}^{n+1}$.

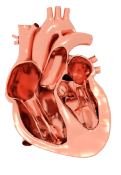
Advection

First we account for the $\mathbf{u} \cdot \nabla \mathbf{u}$ term by convecting $\mathbf{u}^n$, the current, incompressible velocity field at time n, forward in time with the FLIP method [32], yielding an intermediate, convected velocity field $\mathbf{u}^*$. Thus, the advection *Partial Differential Equation* (PDE) $\partial \mathbf{u}/\partial t = -\mathbf{u} \cdot \nabla \mathbf{u}$ is discretized in time via

$$\frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} = -\mathbf{u}^n \cdot \nabla \mathbf{u}^n.$$

Pressure

Then, for the pressure term, the intermediate velocity field given by the advection step, $\mathbf{u}^*$, is made incompressible using the method by Batty et al. [34], while also taking into account boundary conditions. The method applies Chorin's projection method that is based on the Helmholtz-Hodge decomposition, which states that a smooth velocity field can be decomposed into a curl-free component and a divergence-free vector component. Chorin's projection method computes the curl-free component of the velocity field and subtracts it from the original velocity field to yield the divergence-free field. To achieve this, we require the unknown scalar



pressure field p by solving the following Poisson equation:

$$\nabla^2 p = \frac{\rho}{\Delta t} \nabla \cdot \mathbf{u}, \tag{2.9}$$

We obtain the pressure by solving the following (sparse) linear system using an (incomplete) Cholesky-preconditioned Conjugate Gradient solver [33, 34, 36]:

$$\frac{\Delta t}{\rho^2} \mathbf{G}^T \mathbf{M} \mathbf{G} \mathbf{p}^{n+1} = \frac{1}{\rho} \mathbf{G}^T \mathbf{M} \mathbf{u}^*, \tag{2.10}$$

with $\mathbf{G}$ a finite-difference gradient operator, $\mathbf{M}$ the diagonal matrix of all fluid-cell masses and $\mathbf{p}^{n+1}$ the vector of unknown pressure values. Once the pressure field is obtained, it is applied to compute the velocity field at time $\mathbf{u}^{n+1}$ from the intermediate velocity field $\mathbf{u}^*$ via

$$\mathbf{u}^{n+1} = \mathbf{u}^* - \frac{\Delta t}{\rho} \nabla p. \tag{2.11}$$

That is, the pressure is projected as follows:

$$\mathbf{u}^{n+1} = \mathbf{u}^* - \frac{\Delta t}{\rho} \mathbf{G} \mathbf{p}^{n+1}. \tag{2.12}$$

With this our model is complete as we have computed how the given velocity field $\mathbf{u}^n$ changes to $\mathbf{u}^{n+1}$ within one time step.

Marker and Cell (MAC) grid

Since we use the FLIP, a hybrid, Lagrangian-Eulerian approach, we use both particles and a MAC-grid to achieve good preservation of small-scale features and low numerical dissipation. Within FLIP, particles constitute the fundamental representation of the fluid and are used for advection. Each particle has a position and velocity vector. For the pressure computation, the pressure and velocity is discretized on a so-called staggered *Marker-And-Cell* (MAC) grid, as shown for a grid cell with index (i, j, k) in Figure 2.10. The velocity of every particle is mapped onto the grid based on the distance to the grid point. A MAC grid makes the computations of the pressure both more accurate and more stable compared to simpler grid structures due to the way the information is stored. When using a MAC-grid the pressure scalar is stored at the cell centre, while the velocity components u, v and w on the other hand are stored on the cell faces, see Figure 2.10. By doing so, the centred finite differences of the velocity can be computed for the cell centres with a higher accuracy. This results in more accurate input for the pressure solver. After the pressure solve, the velocity of the particles is updated based on the grid. For more details on the implementation of the FLIP method see Section 6.1 and Zhu et al. [33].

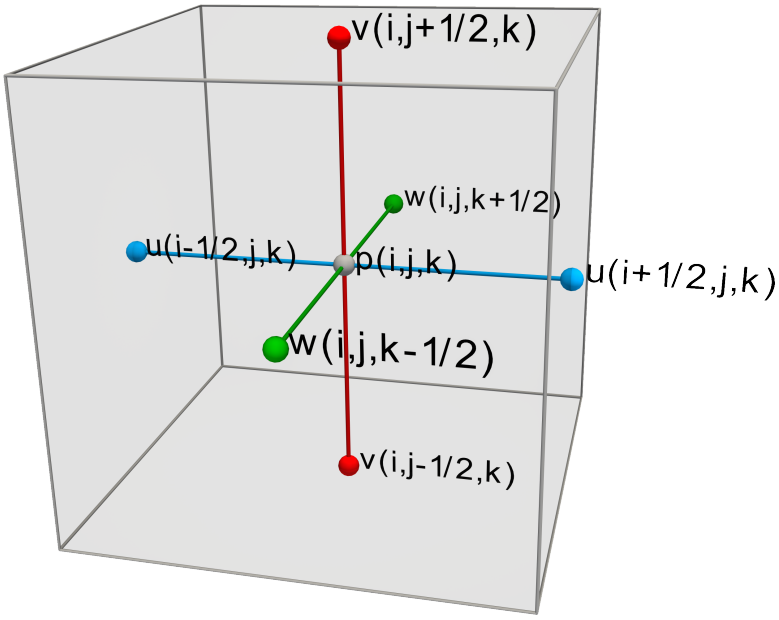
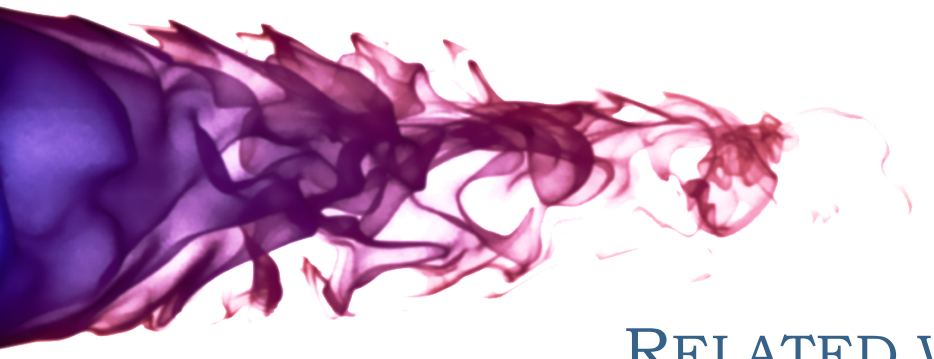


Figure 2.10: The MAC grid stores the pressure in the cell centre, while the velocity components u, v and w are stored on the cell faces.

2.5. Conclusion

In this chapter, the background knowledge required for the understanding of this work were given. Parts of the cardiovascular system and some of the diseases that can occur within the cardiovascular system were introduced and will be referenced in later chapters. Since this work is on the blood flow within the cardiovascular system, the flow data and its acquisition was discussed, with a focus on the PC-MRI data that is used in all remaining chapters. Finally, the concept of CFD, e.g. the simulation of fluids, was presented. In both Chapters 6 and 7 such simulations are used in combination with PC-MRI measurements.



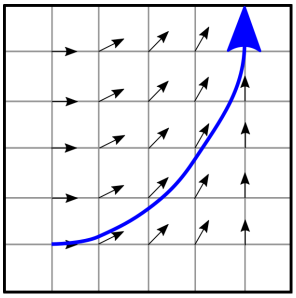
3 Related work

In this chapter, an overview is given of previous work that is relevant for this dissertation. The work presented in this thesis is related to different research fields. The field of flow visualization is introduced and the current state-of-the-art is presented with a focus on blood flow visualization. Then, various approaches dealing with noise in *Phase-Contrast Magnetic Resonance Imaging* (PC-MRI) data are introduced. This includes techniques for the visualization of such noise as well as denoising approaches. Finally, the field of data assimilation is presented with a focus on PC-MRI measured data. Each of these topics are discussed in their own section.

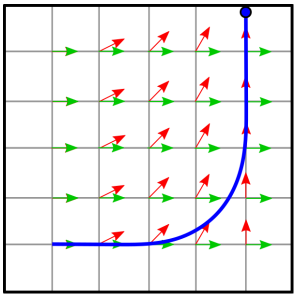
3.1. Flow visualization

The visualization of blood flow is closely related to the visualization of flow in general. Such visualizations are relevant for many disciplines, e.g. engineering, and the same basic techniques are being used. Flow in general can be characterised in two types: *steady* and *unsteady* flow. Here, steady refers to a flow that is not specified by a fixed moment in time, since the flow does not change over time. That is, the flow is defined by a single velocity field. Whereas unsteady flow, sometimes referred to as transient flow, is time dependent and consists of a velocity field that changes through time. Moreover, flow data can be two or three dimensional. Naturally, when visualizing the flow, 2D flow suffers less from occlusion and clutter compared to 3D flow.

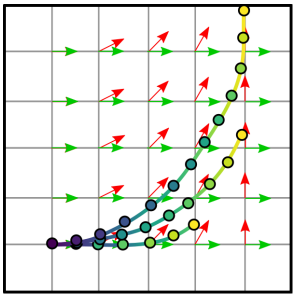
To show flow physically, *streamlines*, *pathlines* and *streaklines* are common approaches. For example, to visualize the flow of water, often a tiny drop of dye or ink is dropped in the water. The location at which the droplets are released is called the *seeding location*. In case of a steady flow, the drop of dye or ink will have an identical trajectory to a drop that is released later. Hence, a streamline is a line where the next point on the line is determined by the direction of a velocity field at that position, as Figure 3.1a shows. For unsteady flow, pathlines and streaklines



(a) Streamline visualization. The line visualizes the trajectory of a massless particle through the underlying steady velocity field. The arrow glyphs in essence represent short streamlines as well.



(b) Pathline visualization. The line represents the trajectory of a single particle over time. The direction of the path is determined by the velocity field at the given moment in time.



(c) Streakline visualization. The line consists over multiple particles released continuously over time, hence the line changes over time as indicated by the three lines and the color of each particles.

Figure 3.1: Stream-, path- and streakline visualizations. For the time-dependent visualizations, i.e. path- and streaklines, the velocity field linearly changes direction over time from the green arrows to the red arrows. Each of these visualization shows different aspects of the data and has their own advantages and disadvantages.

also exist. Similarly to a streamline, a streaklines is generated using a continuous stream of dye or ink being released in the fluid which then forms a line. As a result, a streakline can change over time, as shown by shown by Figure 3.1c. A pathline on the other hand is determined by the trajectory of a single droplet. As such, a pathline cannot be visualized physically directly since the droplet will be a single point. To visualize a line from this single droplet additional techniques are needed, for example using a camera with a long exposure time, as illustrated by shown by Figure 3.1b. Note that, pathline and streakline can be used in the case of steady flow, however, the resulting visualization will be identical to a streamline.

Streamlines, pathlines and streaklines can also be generated computationally by integration of a vector field. For this, virtual massless particles (virtual ink or dye) can be released for the visualization of flow [2, 38]. The particle is usually seeded from a user defined position that marks the beginning of the trajectory of the particle. Note that, it is possible use streamlines for a single measurement in time. For blood-flow visualization, streamlines and pathlines are often used by domain experts as these provide an static overview of the flow and are most often used to communicate findings [2, 11, 17–19, 39]. If many lines are shown to the user, clutter and occlusion. To avoid this, line visualizations are often filtered to reduce the number of lines that are shown to the user and to produce an effective representation of the flow [40, 41]. Multiple frameworks for such a visual exploration for PC-MRI data exist [21]. Streaklines are typically not used for the visualization of blood due to requiring more computer memory to be available in comparison to stream- and pathlines. However, they allow the visualization of flow patterns that change over time [42].

Besides qualitative analyses, comprehensive flow visualization is essential to

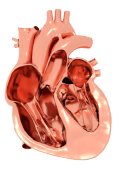
Figure 3.2: LIC applied on a vessel structure. A noise texture is advected over the vessel wall to represent the flow field near the wall.

gain understanding of the intricate (blood-)flow patterns in relation to disease development [43]. For this computers can be used to generate flow visualizations based on measured or simulated flow data. Since for some locations the velocity vector is known, a visualization could show this using a so-called *glyph*. These glyphs are symbols that can indicate the local properties of a flow. A common glyph used for flow is an arrow, where the orientation of the arrow corresponds with the direction of the flow and the length of the arrow shows the speed of the flow. However, only a limited amount of such glyphs can be used to avoid visual clutter to the point where the observe cannot identify individual glyphs anymore.

Another general flow visualization approach is visualizing the massless particles themselves. In computer-based flow-visualization approaches, experimental visualizations are often mimicked using particles. Surface particles [44] (i.e., particles with a normal vector based on the surface of the particle defined by the flow) can be used to show the direction of the flow. Another method is to use semi-transparent textured billboards, where the density of the particles is proportional to the opacity [45]. More advanced techniques use smoke surfaces [46] reconstructed from a high number of particles advected through the flow. To the best of our knowledge, these methods were not yet applied for the visualization of blood-flow data, however, they are able to identify flow patterns, such as vortex creation, that are difficult to obtain by other visualization methods.

Instead of lines and particles, also texture-based methods are sometimes used. Here a texture is mapped to the flow and deformed by the flow, a commonly used texture-based approach is *Line Integral Convolution* (LIC). LIC uses a noise texture which is then advected by the flow. Typically it is used for 2D flow or for flow on or near surfaces as shown by Figure 3.2. While it is possible to generate a 3D noise texture, 3D LIC leads in general to occlusion and clutter [47]. This makes LIC challenging to apply for 3D unsteady flow data.

All of these techniques employ numerical methods, e.g., Runge-Kutta integration, to visualize the flow data. Many of such flow visualizations are thus in essence



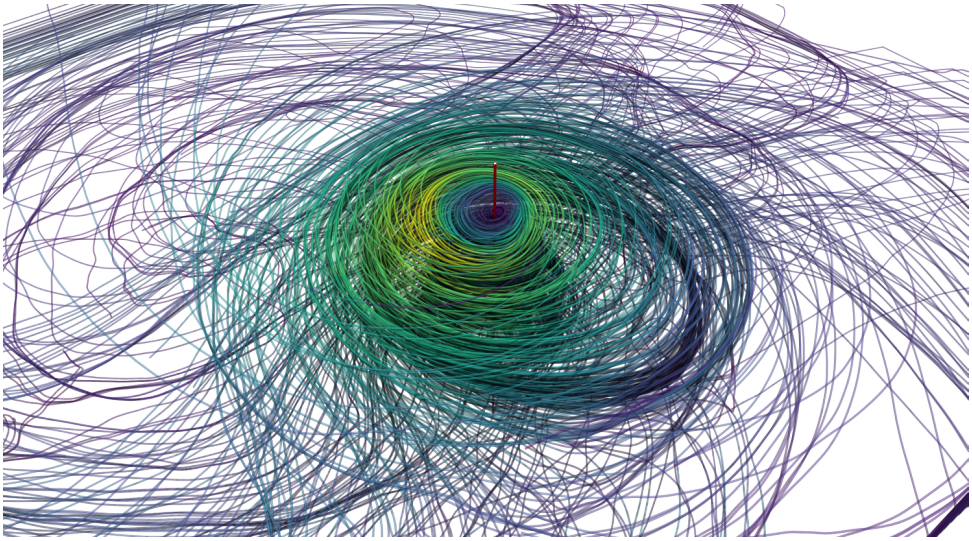


Figure 3.3: A streamline visualization of the air flow within hurricane Isabel, a category 5 hurricane from September 2003. Hurricanes are massive vortices that rotate around the so-called eye which is indicated by the red bar.

Lagrangian, i.e., massless particles advected through the vector field and use lines [39] or surfaces [48] to visualize the flow. However, in the last years, a focus has been put on visualizing flow features such as vortices [21, 43, 49].

3.1.1. Flow feature extraction

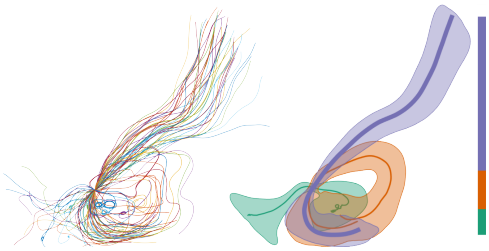


Figure 3.4: The aggregation approach of Ferstl et al. on a ensemble of streamlines. The representative streamlines are shown in the middle of an abstract visualization of the identified major patterns in the ensemble. Figure from the work by Ferstl et al. [50] © 2016 IEEE.

For an effective visualization, not only integral lines are used, as they only provide a limited view on the flow. Direct visualization of certain features or patterns that occur the flow is often more relevant. Offering a user visualizations of such patterns in the flow that are of interest can reduce the amount of analysis. Some flow features that can be of interest are vortices, helices, turbulence and reflux, e.g. backwards flow. Such features can be computed from the flow and can provide more quantitative information or indicate interesting flow patterns. It can also reduce occlusion and clutter. One particular flow pattern that has received much attention in cardiac flow is the vortex. A vortex can be described as fluid flowing in a circular motion around a certain axis. Examples of well known natural occurrences of vortical flow include

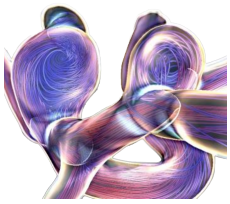


Figure 3.5: Segmentation-based context visualization using clipping. Figure from the work by Lawonn et al. [49] © 2014 Wiley.

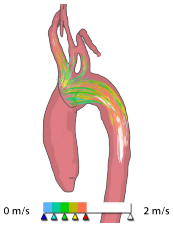


Figure 3.6: Illustrative rendering of the segmentation with the flow rendered on top. Figure from the work by Pelt et al. [39] © 2010 IEEE.

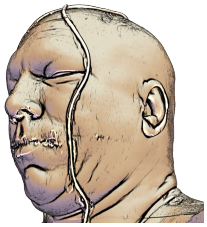


Figure 3.7: Illustrative rendering of volume data using contour lines. Figure from the work by Kindlmann et al. [61] © 2003 IEEE.

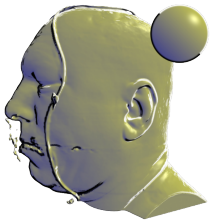


Figure 3.8: Illustrative rendering of volume data using lit sphere maps. Figure from the work by Bruckner et al. [62] © 2007 Wiley.

tornadoes, whirlpools and hurricanes, of which the later is shown in Figure 3.3. However, no formal mathematical definition of a vortex exists for unsteady flow. Naturally, there is a lot of literature within the field of fluid dynamics and *Computational Fluid Dynamics* (CFD) on vortices. Different approaches to identify vortices exist, partly due to the existence of many different definitions of a vortex [41, 51–53]. Within unsteady flow, flow features can move, form and disappear of over time. These moments of creation and destruction of vortices, as well as the area encompassed at each time step in between, can be visualized on a timeline [54, 55]. Both Stalling et al. [56] and Doleisch et al. [57] worked on visualization based on features. Their approaches rely on the user to provide a mathematical feature of interest. By computing where the mathematical condition holds, the seeding areas for the visualization of these features can be generated.

3.1.2. Clustering

Besides using feature extraction one could also use clustering of flow lines to provide the user with an overview of the flow. For example, the work by van Pelt et al. [58] proposes a clustering of the flow lines based on their spatiotemporal aspects. By showing representatives of the cluster, the amount of lines shown to the user is reduced leading to an image with less clutter and occlusion. Similarly, the work by Ferstl et al. [50] uses clustering to visualize the representative of an ensemble of multiple flow fields, as shown by Figure 3.4. Moreover, by clustering flow lines that are similar, it is possible to not only generate representative flow lines, but also to classify different flow patterns [59, 60].

3.1.3. Anatomical context visualization

Besides the visualization of the flow, understanding the context in which it takes place is very relevant. Hence, showing the anatomy of the vessel in combination with the flow may be beneficial. To be precise, the context is helpful for the evaluation of the flow, localization within the body and the relation of flow features to the morphology. Many different approaches for such a visualization exist.

Rendering

The context should also be shown to the user in a meaningful way. Often vessel surfaces are rendered that show anatomical context visualization [39, 49, 63]. To avoid occlusion the measured dataset can be shown on the background or a plane in a 3D visualization [64, 65] or by clipping of the rendering using planes or boxes and other voxel-based clipping volumes [66]. Furthermore, clipping is done based on the visualization and the viewing angle [49], as shown in Figure 3.5. Illustrative techniques are used to emphasize the focus of a visualization, and abstract from unnecessary details [67]. Using few colors in the shading of the context visualization can make it less distracting while still providing enough perception of its structure [39], as shown by Figure 3.6. Contour lines, shown by Figure 3.7, were introduced to emphasize this structure even more [61]. A different approach was proposed by Bruckner et al. [62] that allows the user to select and create different illustrative styles. It allows for various transfer functions, that interpolate between different illustrative styles, called lit sphere maps [68], instead of colors, as shown by Figure 3.8.

Segmentation

To visualize the context additional data has to be provided or generated. For this, often segmentation is used on image data, Such a segmentation divides an image into different segments each with their own meaning, e.g. in and outside the vessel. Several forms of segmentation exist from manual segmentation, automatic segmentation [69, 70], to user-guided automatic segmentation [71, 72]. For these segmentations to work well, much higher quality anatomical data compared to the available magnitude obtained from the PC-MRI data should be used.

Most of the visualization frameworks for PC-MRI, however, focus mainly on the exploration of vascular flow instead of flow in the heart. The ventricles are often an area of interest when inspecting cardiac flow. While many different methods for automatic segmentation of the ventricles exist [73–75], all of them require higher quality anatomical data to be available. Pelt et al. [48] developed a virtual probe that can be placed manually or fitted to a vessel, defining a volume of interest where we want to inspect flow. This probe is intended for vessels and the fitting requires laminar flow. The probe's shape is a truncated cone, which fits the tubular structure of vessels well. For the left ventricle however, a half-ellipsoid shape provides a better approximation [76].

3.2. Noise in PC-MRI

As with every measurement PC-MRI suffers from uncertainty in the form of noise and artefacts. In this section various approaches are discussed to effectively deal with such uncertainty.

3.2.1. Visualization

Given a model of the uncertainties present in the data such as presented in Section 2.2.3, effective communication of this uncertainty to the end user is a chal-

lenge. Many uncertainty visualization methods rely on visual cues using the available graphical attributes, such as color, size, transparency, position, texture [77], or fuzziness [78, 79]. Other methods add geometrical objects, employ animation [80], sound, or haptics and interaction [81].

For vector fields, uncertain variations caused by numerical integration were previously visualized by means of uncertainty ribbons, path trajectory envelopes, and so-called twirling batons [82]. Ferstl et al. [50] and He et al. [83] both propose methods of aggregation to visualize ensembles of path- or streamlines by identifying representatives. Ferstl et al. includes a surface visualization to represent the the ensemble of lines, as shown by Figure 3.4. However, a single surface does not capture the density of the path- or streamlines well.

Several uncertainty visualization methods for vector fields have been proposed. Uncertainty glyphs is a common choice [84, 85]. However, glyphs tend to suffer from clutter and occlusion in 3D limiting what can be shown. Extensions of two-dimensional LIC to include uncertainty where introduced by Botchem et al. [86] and Otto et al. [87]. Bhatia et al. [88] introduce *edge maps* a LIC-based visualization to indicate the uncertainty involved in computing streamlines and topological structures. Unfortunately, as mentioned before, in general LIC leads to occlusion and clutter in 3D [47] which makes it difficult to apply for 4D PC-MRI data.

Visualization of uncertainty in flow feature based visualizations has also been introduced. Petz et al. [89] and Otto et al. [90] derived 3D uncertainty scalar fields that then were visualized with volume rendering approaches. Guo et al. [91] introduce a method similar to LIC to cope with uncertain unsteady 2D or thin 3D vector fields. While the results are promising, the method does not perform at interactive rates. Furthermore, the method is non-trivial to extend to volumetric data without introducing occlusion problems. These methods have not been applied to PC-MRI data. In this thesis, however, we focus on the visualization based on integration lines commonly used in this context.

A first attempt to include uncertainty information in PC-MRI visualizations was presented by Friman et al. [24]. They have modelled the noise in measured blood-flow unsteady data as a multivariate Gaussian distribution discussed in Section 2.2.3. In their work, they present the uncertainty information using a flow map. They visualize and quantify the uncertainty using conventional flow visualization techniques, such as streamlines and par-

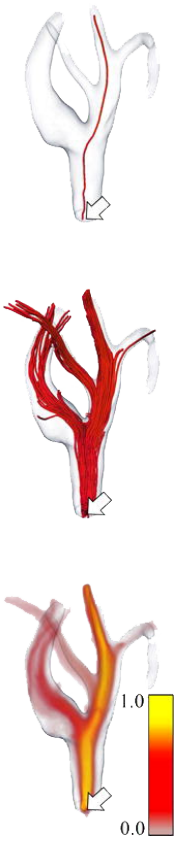


Figure 3.9: From top to bottom a traditional pathline visualization, the corresponding probabilistic pathlines and the uncertainty distribution of the pathlines. Figure from the work by Friman et al. [25][1].

[1]Reprinted from Medical Image Analysis, 15, O. Friman, A. Hennemuth, A. Harloff, J. Bock, M. Markl, and H. Peitgen, Probabilistic 4D blood flow tracking and uncertainty estimation, 720 - 729, Copyright (2011), with permission from Elsevier

ticle traces which often suffer from occlusion. The extension of their work includes pathlines and particle traces that show the probability distribution [25], providing a quantitative measure for uncertainty, as shown by Figure 3.9. Both works by Friman et al. [24, 25] rely on sequential Monte Carlo sampling of the probability space of the 4D PC-MRI data, which is computationally expensive, hampering interaction. The visualizations proposed by Friman either summarize independent pathlines without indicating the trajectories or only shows pathlines for one seeding position. In a follow-up work by Schwenke et al. [92], the uncertainty was represented by the likeliness of the trajectories using a fast marching method. However, the technique by Schwenke et al. is not yet applicable to time-varying data and is non-trivial to extend to unsteady flow, and thus can only capture streamlines.

For derived flow features and their uncertainty, the amount of research is still very limited. For the computation of the stroke volume and regurgitation fractions, important indicators of the effectiveness of the flow, the method by Köhler et al. [93] takes the uncertainty of the measured data into account to derive robust measures that are insensitive to the exact angulation of the measuring plane. Our method is complementary to the approach by Köhler et al. and could be used as an extra visualization in their context.

Despite the various methods for visualizing the uncertainty of flow that already exist, most of these methods are not readily suitable for the visualization of the uncertainty of 3D PC-MRI measurements over time at interactive rates. One of the main challenges of these methods is the occlusion and often the relation of the visualization to its seeding position is lost, making it more difficult to track the flow.

3.2.2. Denoising

It is, apart from its visualization, in general, interesting to remove noise from data whenever possible. For this so-called *denoising* multiple approaches exist specifically for PC-MRI data [21, 37, 94–100]. Many of these methods try to reduce the noise by making the flow physically more plausible by removing divergence. The reasoning is that the flow does not have divergence, so any divergence present is caused due to measurement imperfections. Some of these methods include the *Finite Difference Method* (FDM) [94], which projects the data onto a divergence-free vector field. Another method uses *Radial Basis Functions* (RBF) [98] to minimize the divergence approaching the measured data using a set of convolution and divergence-free radial basis functions. Similarly, Ong et al. [100] proposed the use of *Divergence-Free Wavelets* (DFW). A comparison by Sereno et al. [37], however, found that some divergence remained present in the data for the methods above. They reduce its divergence, but there is no control to have resulting data that is closed to the original measured data. Some methods actively try to minimize the difference between the outcome and the noisy input data, e.g., Bostan et al. [99]. However, these techniques mainly focus on making the data divergence free, and they do not consider the vessel walls. Furthermore, most of these methods rely on the Helmholtz-Hodge decomposition, as described in Chapter 2.4.1 to make the data divergence free. However, this decomposition assumes that the input vector

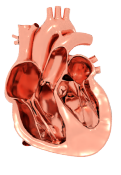
field is sufficiently smooth [101], which is not the case for measured data.

3.3. Data assimilation

One of the goals of this thesis is to use a physics-based model and measured data to obtain physically-plausible patient-specific data. For example, Le et al. [102] use simulation models guided by PC-MRI data, i.e., the boundary conditions are derived from measured PC-MRI data. In this case, however, only a small portion of the measured flow data is used. The concept of using modelled and measured data to obtain better data is called *data assimilation*. More precisely, data assimilation is the process of combining observed (i.e., measured) data of a system with scientific information, typically a mathematical model of the system, to obtain an estimate of the true state of the system. It attempts to use all available information of a system to estimate its true state. Hence, it can be used for denoising, interpolation and extrapolation of measured data. Multiple data assimilation techniques exist and are being used in many fields, such as meteorology [103], geoscience [104], climatology [105] and numerical weather prediction [106–108]. A commonly used data assimilation method is *3-Dimensional Variational Assimilation* (3D-VAR), which minimizes the weighted squared difference between the observation and model.

In computer graphics, having control over simulated fluid is an active research field. The main objective there is to provide artists with tools and methods to control the fluid simulation to achieve a desired animation of the fluid. Such controllability is often employed by animators to produce various effects [109]. For example by allowing the artist to use reference images of the flow density [110], control forces [111, 112], control particles [113] or, more recently, data-driven techniques [114]. Another approach was proposed by McNamara et al. [115] and uses iterative minimization to match a given target density or surface. A variational approach was introduced by Nielsen et al. [116] where the simulation follows the low frequency information present in the target flow. Note that, for animation, the *appearance* of the fluid is most important to the artist, that is, the fluid surface in the case of liquids and densities in the case of smoke. Therefore, these control methods try to control the fluid surface or density. Since, in this work, we do not focus on the visual representation of the flow but rather on the underlying velocity field, therefore, the existing methods cannot readily be applied to our scenario.

In the medical domain, some techniques exist that use data assimilation to match simulation data with measured data or use the data to evaluate the simulation outcome [117]. Many of these approaches focus on finding the optimal boundary conditions to match a target velocity field [118–122]. They consider the whole measurement to find the in- and out-flow conditions that best match the measured data. A disadvantage of this approach is that it is strongly dependent on the accuracy of the modelled anatomy. Updating the in/out flow conditions cannot generate flow patterns if the anatomical segmentation does not have the correct details. One approach that couples the full imaging data with a simulation was proposed by Funamoto et al. [123]. The authors aim to reduce acquisition artefacts by integrating 2D Doppler ultrasound data with a fluid simulation, steering the re-



sults with a feedback signal based on the difference between the measurement and simulation. Data assimilation has been used in a similar context before [119, 124–126]. However, none of these methods are applied to 4D PC-MRI measurements directly, and hence, do not provide a 3D flow field over time while being minimal invasive for the patient. The approaches by Rispoli et al. [127] and Fathi et al. [128], on the other hand, use regularization to compute a least squares solution to match model and measured PC-MRI data based on the actual full 3D flow in the patient. However, they do not consider the uncertainty in the measured data nor allow for temporal interpolation. In this work, we present data assimilation approaches using the full 4D PC-MRI measured data that focuses on denoising and spatial and temporal interpolation. Our goal is to provide physically-plausible data that is close to the patient-specific measured data.

4

General Blood-Flow Visualization

Typically Phase-Contrast Magnetic Resonance Imaging (PC-MRI) data is visualized using stream- or pathlines. However, time-varying aspects of the data, e.g., vortex shedding, breakdown, and formation, are not sufficiently captured by these visualization techniques. Experimental flow visualization techniques introduce a visible medium, like smoke or dye, to visualize flow aspects including time-varying aspects. In this chapter, a framework is proposed that mimics such experimental flow visualization techniques by using a high number of particles. The framework offers great flexibility which allows for various visualization approaches. These include common traditional flow visualizations, but also streak visualizations to show the temporal aspects, and uncertainty visualizations. Furthermore, various approaches are presented to mitigate any clutter and occlusion, that is often present in 3D flow visualization, for example through glyph-like visualization. The framework has been adopted by domain experts to visualize the vortices present in the sinuses of the aorta root showing the potential of the framework. Furthermore, an evaluation among domain experts indicated that having the option to visualize the uncertainty contributed to their confidence on the analysis.

This chapter is based on the following publications:
InkVis: A High-Particle-Count Approach for Visualization of Phase-Contrast Magnetic Resonance Imaging Data by N.H.L.C. de Hoon, K. Lawonn, A.C. Jalba, E. Eisemann and A. Vilanova in *Eurographics Workshop on Visual Computing for Biology and Medicine (2019)* [129] © 2019 Eurographics
Presence of aortic root vortex formation after TAVI with CENTERA confirmed using 4D-flow magnetic resonance imaging by J. Vendrik, E.S. Farag, N.H.L.C. de Hoon, J. Kluin and J. Baan Jr [19], reprinted by permission from Springer Nature, International Journal of Cardiovascular Imaging © 2018.

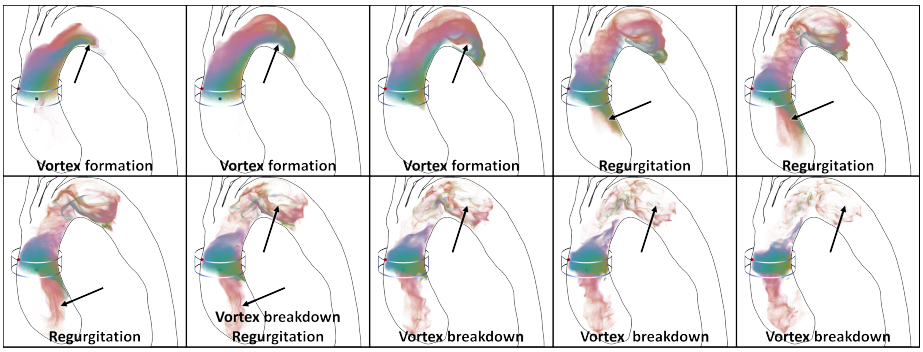


Figure 4.1: A streak visualization showing the formation, shedding and breakdown of a vortex in a patient with an aortic dissection in the aortic arch and regurgitation is present in the ascending aorta.

As indicated by Section 3.1 visualization of the blood flow is an active field of research. The observed patterns in a patient's blood-flow determine the probability of a disease, and, ultimately, can establish a final diagnosis. For example, for certain CVDs, vortex-flow patterns are considered to be an essential factor in the development of these diseases [17, 41, 130, 131]. Therefore, visualization of the flow is important, such that it is possible to locate and qualitatively analyse the flow features of interest [17, 38, 130]. Often pathlines or streamlines are used to visualize the blood flow [2, 11, 17, 18, 39]. However, these visualizations do not capture any of the time-varying aspects of the data, such as, how and when vortices form and breakdown, and how they move through the flow (shedding), i.e., information on the evolution of a vortex over time. Which is, for example, important when analysing the flow in the heart [131]. An example of vortex formation and breakdown in the aorta with the framework is shown in Fig 4.1. While vortices can be visualized using pathlines, they cannot be used to visually show how they change over time. The pathlines show a static representation of the temporal behaviour of particles, since they only show a single trajectory for a given seeding position, not how this trajectory can change over time. Streaklines, on the other hand, are more adequate to reveal time-varying flow behaviour [42]. To obtain streaklines in physical flows, visible foreign material is continuously added. For example, in the medical setting, an angiographic catheter are used to inject contrast dye into the artery to evaluate the flow inside it [132]. By continuously seeding particles from a fixed seeding position, the temporal relation between the particles is maintained and flow changes over time are visually encapsulated. Hence, using streaklines one can recognize both local and global spatial and temporal changes in the flow, that cannot be directly provided by streamlines nor pathlines.

One of the reasons streak visualization is rarely applied for the computer-based visualization of blood flow is that it typically has a higher computer memory footprint, i.e., more temporal data must be available to the visualization. To the best of our knowledge, no system uses streak visualization for PC-MRI data.

Moreover, when streamlines or pathlines are used, finding the right moment and

location for the seeding of the visualization is crucial and non-trivial and choosing the wrong settings can lead to missing important flow features. One common approach is to place a fixed number of random seeds over space and time that cover the whole vessel. A disadvantage of this approach is that it leads to clutter if the number of seeds becomes high. While clutter can be reduced by reducing the opacity based on the velocity, e.g. making low velocity flow more transparent, important flow features, such as vortices, will be less visible as these often have a lower velocity [5]. Furthermore, in order to understand the temporal behaviour of the flow the user has to have a mental map of the various moments in time that are presented. In general this method of seeding is considered non-optimal [133]. Because streak visualization requires a continuous seeding over time, the user gets a complete overview of the flow along with its temporal behaviour. Therefore, streak visualization can be used minimizing the demands on the adequate seed-point definition. Yet, effective 3D flow visualization over time presents challenges due to clutter and limited visual channels. Hence, this chapter also presents various strategies implemented in the framework to address these issues, for example a glyph-like visualization.

Another aspect of *Phase-Contrast Magnetic Resonance Imaging* (PC-MRI) data that is often not considered in visualization is the presence of several sources of uncertainty, e.g., phase-wrap artefacts, motion artefacts, partial volume effects and measurement noise for example due to inhomogeneity of the magnetic fields. Using traditional flow visualizations, the users have no easy means to assess the effect of sources of uncertainty in their data. Uncertainty analysis is recognized as an essential stage in any decision making process. Professional analysts and health-care professionals will take regular recourse to uncertainty analysis [134]. They often use their experience to take statistics and uncertainties into account for decision making [135]. However, for complex data where the pipeline from the original data to the visualization is highly complex considering uncertainty on the decision making process becomes very challenging. As a result, uncertainty is ignored, which can be misleading and generating a false sense of reliability, or it can even cause distrust and conflict.

For example, showing a single pathline does not consider its randomness due to that each voxel rather contains a probability distribution of the velocity vector, e.g., due to measurement noise, than a single velocity vector. Furthermore, it is known that the presence of measurement noise has a significant influence on the quantification of the vorticity of the flow [136]. The visualization of uncertainty is recognized as one of the key challenges in flow visualization [137–139]. The flexibility of the framework allows for an interactive uncertainty visualization based on a per-voxel probability distribution, and therefore, presents the user with a way to assess the uncertainty of the flow visualization. Furthermore, we provide user interaction methods to facilitate this exploration, and reduce through guided filtering the extra clutter that visualizing the uncertainty supposes.

In the presented framework, the measured data is sampled and distorted based on the statistical noise model of Friman et al. [24, 25]. The modeling of uncertainty is considered out of the scope of this dissertation. Our work is based on the as-

sumption that a statistical model of the uncertainty per volumetric position exists. Notice that this excludes the uncertainty in the seeding position, and therefore, despite its importance [140], is considered out of scope of this dissertation.

In this chapter, a framework is presented for the visualization of PC-MRI data using a high number of mass-less semi-transparent particles. We apply GPU-based particles that are are traced and released in parallel to be used in the context of PC-MRI data. The flexibility of this framework allows for various visualization approaches. These include common traditional flow visualizations, and adds the possibility for streak visualizations and uncertainty visualizations.

4.1. Mimicking Experimental flow

The presented framework is inspired by experimental flow visualization, which often works by injecting a foreign material as visual medium into the flow, for example dye or smoke, to create a visual representation of the transport of the material by the flow. The foreign material typically consist of fluid with properties similar to the fluid that is to be inspected.

In computer visualization, we can use a high number of virtual particles to mimic such experimental flow visualizations. By using the *Graphics Processing Unit* (GPU), a high amount of particles can be advected in parallel and shown in real time. An efficient implementation of particle advection helps to achieve interactive frame rates [141, 142]. While such particle systems already exist, some components have to be adapted to accommodate for use with PC-MRI data. In this Section more details on the particle system are given.

4.1.1. Integration

In our system the particles are advected in parallel through the velocity field using a fourth-order Runge-Kutta *Ordinary Differential Equation* (ODE) solver on the GPU. By using the GPU every particles can be advected in parallel, resulting in interactive frame rates, even for a high number of particles [141, 142]. The time step is bounded by the *Courant-Friedrichs-Lewy* (CFL) condition that ensures that a particle moves at most one voxel per time step, Δt, to reduce the numerical error, i.e.,

$$\Delta t = \frac{voxelSize}{v_{enc}}. \tag{4.1}$$

Here Δt is the maximum safe time step in seconds, $voxelSize$ is the smallest dimension of the voxels in meters, 0.002 meters (2mm) in our data sets, and v_{enc} is an acquisition parameter (in m/s), representing the largest speed that can be measured unambiguously.

By allowing integration over time and varying the seeding time of the particles, our framework can generate stream- path- and streaklines, as shown in Figure 4.2, with, from left to right, these three integration options. The stream visualization fails to capture the vortex present in the aortic arch of a patient with a aortic dissection. The path visualization fails to show the regurgitation, however, it clearly shows the

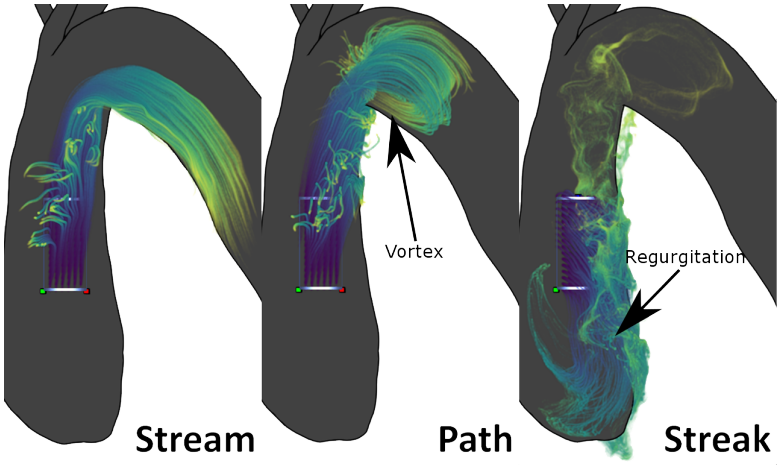


Figure 4.2: From left to right to three integration options are shown, i.e., stream, path and streak integration. All three methods show different aspects of the flow. The color brightness indicates the age of the particles.

vortex in the arch. The streak visualization does show both the regurgitation in the ascending aorta and the vortex in the aortic arch in a single image, however, it results in a more complex image.

4.1.2. Seeding strategies

In both experimental flow visualization and computer-based flow visualization the seeding positions are a crucial step, which is commonly user-defined. Flow patterns can be missed due to too many seed positions, or insufficiently represented due to too few seed positions or seeding at the wrong position. We allow the user to interactively set regions for seeding the visualization and the number of particles that are seeded per region. The particles are randomly seeded in this area to avoid structured artefacts [39].

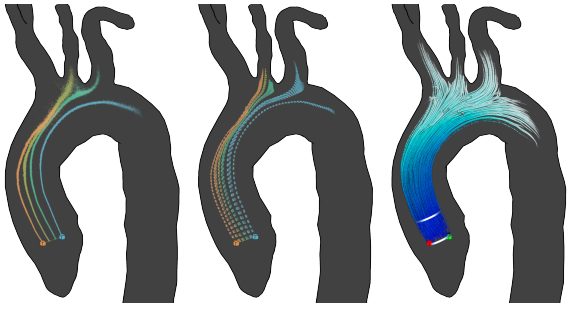


Figure 4.3: By providing different seeding strategies various flow visualizations can be achieved. In this case a stream visualization. From left to right, an uncertainty visualization with discontinuity on the spatial seeding location, a visualization with discontinuity in both spatial and temporal seeding location, and a volumetric seeding region where particles are seeded from each voxel centre.

We also propose to take advantage of the structural artefacts, e.g., by seeding

repetitively from a fixed position, it is possible to generate line-like visualizations, given the high number of the particles [141, 142]. For example, if a single point is used for injecting dye of a given color, the dye will show the flow originating from this point by means of a line, i.e., a streakline [132]. By allowing control on the discontinuity of seeding both on the spatial and temporal seeding location, we can obtain images that are similar to experimental flow visualizations. Some examples of different seeding strategies are shown in Figure 4.3.

4.2. Visual representation

In this Section, we describe several strategies that are present in our framework and illustrate its flexibility. Rendering a large amount of particles despite providing flexibility also tends to cause visual clutter. Therefore, we also provide several options to reduce the visual clutter.

4.2.1. Transparency

Transparency helps mimicking smoke and dye, commonly used in physically-based animation. This also provides some intuitive reference in the analysis. To define the transparency falloff from the particle's centre, different kernels could be used, however, using a Gaussian kernel yields an ink-like visualization and is used throughout this chapter. The particles are sorted based on their distance to the viewer and are rendered from the back to the front. Note that the particles are represented by points, and, as such, depth sorting can be done for correct transparency by sorting the points based on their distance to the viewer. The sorting and rendering is done on the *Central Processing Unit* (CPU) to enforce a correct rendering order, which would not be guaranteed when the GPU would be used. While the rendering cannot be done in parallel, the system still achieves interactive frame rates.

The transparency can linearly increase depending on the age of the particle: the older the particle, the more transparent it is rendered. Thus, it seems that particles dissolve over time. Furthermore, the user can set the maximum particle age in the system, hence it is possible to see what happens to the particles over multiple heart cycles.

Having semi-transparent particles is helpful. The use of transparency provides a rather global overview of the flow, while an opaque rendering would occlude the structure of the internal flow.

4.2.2. color encoding

Physicians are commonly interested in understanding how the blood flow distributes. Therefore, the user should be able to identify the origin of the particle, to derive the direction and behaviour of the flow. Furthermore, it also helps identifying the patterns present in the flow. To achieve this effect, we encode the seeding position of each particle using a color, mapped to the seeding positions. To evaluate the relationship of the seeding position to the current position of a particle in absolute terms, we encode the seeding position as a categorical attribute, using color maps

with distinctive hues. We propose to use isoluminant color maps, which makes it possible to use luminance for the visualization of other properties [144]. While the perception of transparency does not rely on the luminance [145], the use of both luminance and transparency is not optimal. However, for our purpose, the use of transparency is required to regulate the visual clutter and to visualize the complex flow patterns. Moreover, an isoluminant color scheme does not present an implicit ordering, therefore every seeding position is perceived as equally important.

For most images in this chapter, unless indicated otherwise, an isoluminant color scheme is used; examples are shown in Figure 4.4. These color schemes were derived from the Hue-Chroma-Lightness (HCL) color space [143], a cylindrical transformation of the CIE L*a*b* color space, such that each equal step through the HCL space results in approximately equal perceptual changes in color. The perceived luminance was kept constant for these color schemes.

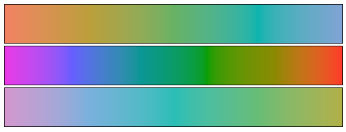


Figure 4.4: Isoluminant color schemes taken from the HCL color space [143].

4.2.3. Depth encoding

It is important to be able to correctly identify the shape and spatial position of flow patterns shown by the particle visualization. It is known that depth encoding is incorporated in our visualizations. To this extent, we included several depth encoding methods, see Figure 4.5, based on the findings of Kersten-Oertel et al. [146]. The conclusion of this study was that aerial perspective [147] and pseudo-chroma depth [148] provide users with the most accurate depth cue, however, they change the underlying color and, thus, the mapping of this color to an underlying value. Furthermore, local differences in depth can be difficult to identify. Kersten-Oertel et al. [146] also indicate that edge enhancement provides a valuable depth cue.

Depth darkening by Luft et al. [149] enhances the edges of visualized objects. Based on the relative distance between two points to the viewer, darkening is applied for the farther point. Depth darkening enhances the edges and thus provides a depth cue that has less impact on the underlying color, however, global depth information can be unclear. This method uses the depth buffer, making it non-trivial to apply on transparent renderings, where multiple layers with different depth maps are visible. In consequence, as we render a particle, we read out the current underlying depth buffer and use the gradient of the depth buffer to decide on the darkness of the particle's halo. This makes it possible to apply depth darkening for transparent particles, and is used for the images in this work. Depth darkening does not interfere with the color of the particles nearest to the viewer and enhances local differences in depth, depth darkening is chosen and used for all the images in this chapter.

4.2.4. Glyph-like visualization

Due to the limited number of available visual channels, it can be helpful to encode certain aspects, for example the speed and direction of the flow, without the use

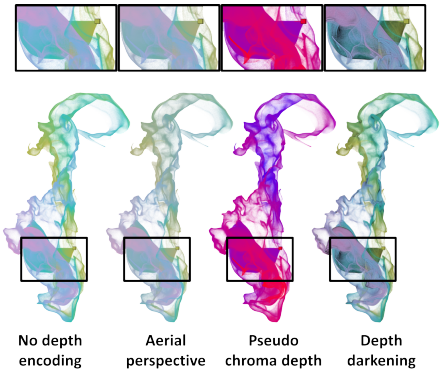


Figure 4.5: Streak visualization of turbulent flow in a patient with a dissection using four different depth encoding options. From left to right: no depth encoding, aerial perspective, pseudo chroma depth and depth darkening.

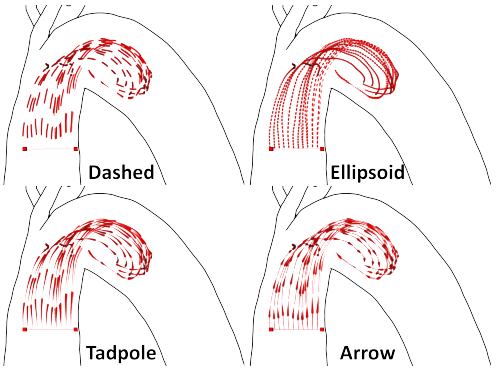


Figure 4.6: By varying the size of the particles that form a cyclical pattern through time, a glyph-like representation can be achieved, indicating the direction and speed of the flow. The particles are seeded one a line.

of color. For traditional line visualizations, the speed and direction is often encoded using textures or glyphs, e.g., an arrow shows both the direction and the speed of the underlying flow without interfering with the color channel. This allows the use of the color channel for encoding different features of the flow. The presented framework, allows the generation of glyph-like visualizations through applying the shape mapping functions introduced by Everts et al. [150]. By varying the size of the particle and seeding through time by defining a cyclic pattern we can vary the particle size based on the particle age to obtain a glyph-like representation, as shown in Figure 4.6. These glyphs-like visualization can be used to provide additional information regarding, for example, velocity and direction, which does not interfere with both color and opacity encoding [150, 151].

4.3. Uncertainty

The flexibility of the framework allows for the visualization of uncertainty of the flow data. In this Chapter, uncertainty is seen as the level of robustness of the information provided, such that the user is aware of the randomness present in the data.

This work does not focus on the derivation of the models that describe these uncertain factors accurately, as this is considered out of scope. Instead, the focus is on the visualization of the uncertainty caused by measurement noise, as most previous work. It is important to note that most uncertainty visualizations will not provide extra information to the expert beyond than a level of confidence on the visualization shown. Hence, quantitative assessment is not the goal of such visualization. To convey uncertainty, so-called fuzzy visualization is a common approach [78, 79]. Such fuzzy visualizations make it harder for the user to distinguish for example specific trajectories if the uncertainty is higher, and hence gives and indi-

cation of the underlying uncertainty.

There is very limited work on showing the effect of the uncertainty in the data. Hence, the goal here is to provide this uncertainty information. We focus on measurement noise that is usually modelled through probability distributions. The flow data resulting from PC-MRI measurements is not necessarily divergence-free. As a result, any particle trace in the data is potentially showing non-physical flow. Hence, the idea is that a trace based on the probabilistic distribution of the noise is as valid as a trace based on the acquired PC-MRI data. These probabilistic distributions expresses the randomness that is present in the scan. Hence, one could say our goal is to visualize the traces that would emerge from multiple scans of the same person and area. Therefore showing the randomness due to acquisition noise. For this, the model by Friman et al. [24, 24] is applied, as presented in Section 2.2.3. Despite giving a limited view, measurement noise adds a source of uncertainty, and the proposed method can be used with other potential models of uncertainty, as long as they are modelled as a distribution that can be sampled.

4.3.1. Distribution sampling

Measurement noise is usually modelled through probability distributions. To visualize uncertainty, we compute random particle trajectories that are defined by sampling the orientation distribution function defined by the uncertainty at each voxel position. These particle trajectory indicate the possible movements that the particles could given an specific measurement or scan considering the acquisition noise. The noise is assumed to be mutually independent at each voxel.

For the computation of the probabilistic particle trajectories the uncertainty distribution has to be sampled. We apply the same approach as commonly used for stochastic fiber tracking [152, 153]. For each integration step, we draw random values from the normal distribution from Eq 2.4 using the Box-Muller approach. By randomizing the velocity for every sample based on the signal strength of the voxel and the variance, the sampling is independent of the time-step size. Note that, the signal strength combines the three velocity directions and the magnitude image, and as such, the added noise depends on all these components. The sampling is done independently for each position. However, neighbouring sampling locations will have a similar signal strength when the underlying velocity in the real flow field strongly depends on neighbouring flow.

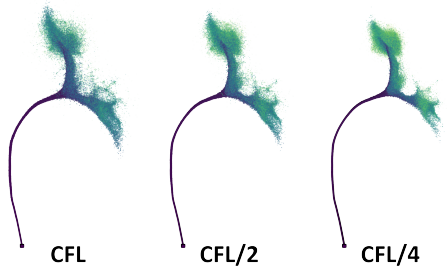


Figure 4.7: The uncertainty is shown using the Viridis color map. Brighter colors indicate a higher cumulative deviation. Left image: the sample step size is given by the CFL condition; centre and right images: the sample step size is set to half and a quarter that by the CFL condition, respectively.

From this, we can also compute the difference $diff$ in m/s between the mean velocity vector ($\mathbf{vel}_m$) and the random velocity vector ($\mathbf{vel}_u$) obtained by random

sampling using the covariance matrix

$$diff = ||\mathbf{vel}_m - \mathbf{vel}_u||. \tag{4.2}$$

By accumulating this local difference for each integration step, weighted by the step size in seconds, we obtain a cumulative deviation. This cumulative deviation indicates how much the particle deviates from the measured flow. Hence, if the particle passes through an unreliable voxel the deviation is more likely to be bigger in comparison to when it passes through a more reliable voxel. With an infinite number of particles and a time step that goes to zero we would obtain the true stochastic distribution for a given seeding position. Therefore, we need a high number of particles and a time step that is as small as possible. The user can control both parameters to fine tune the balance between accuracy and performance. Note that, by using the CFL condition for the sample time step size, it is upper bounded and provides results that are comparable to a much smaller time step sizes especially when the certainty is high, as shown in Figure 4.7 where brighter colors represent a higher cumulative deviation. Using a larger time step, the distribution width is slightly overestimated and the cumulative deviation is underestimated for uncertain regions. The user can, however, choose to have a higher accuracy of the uncertainty distribution at the cost of performance and thus obtain a better approximation of the stochastic trajectory of a particle when needed, as shown in Figure 4.7. Note, however, that the visual differences are minimal, suggesting that the CFL condition is a sufficient approximation.

4.3.2. Uncertainty visual representation

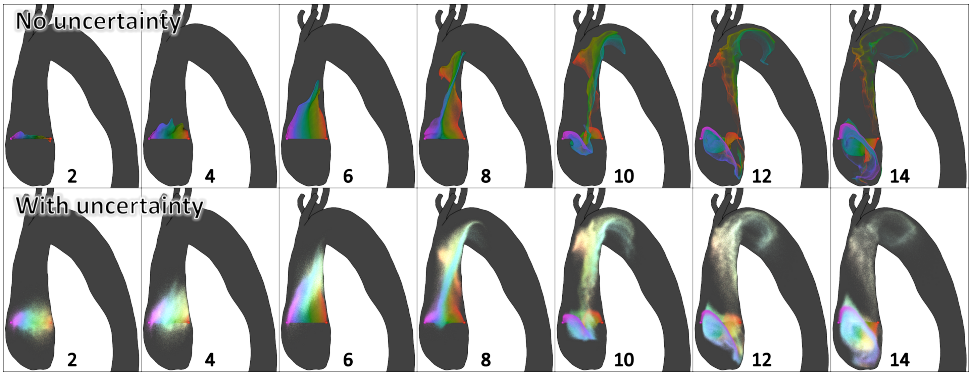


Figure 4.8: Streak visualization of systolic flow in a patient with an aortic dissection. The bottom row uses our uncertainty visualization. The numbers indicate the current phase of the visualization. Particles leaving the mesh were removed.

The particle density in an volumetric area indicates the likeliness of the particle's to arrive to a given location considering the measurement noise and therefore independently of the specific scan. This allows to evaluate the reliability that the particles starting at a given point will reach another area. The use of transparent

particles already indicates the areas of higher density by having higher opacity. However to enhance its value we provide extra visual encodings.

By using isoluminant color schemes, the luminance visual channel can be used to encode the uncertainty. Moreover, the accumulation of luminance indicates a gathering of uncertainty. Note that, if the uncertainty is higher, the visual clutter increases. Additionally, the seeding position becomes difficult to decipher which is in line with that the information is not trustworthy. The relation between the two gives insight in the development of the flow and uncertainty, i.e., the reliability of the seeding position for the flow can be derived.

However, for the investigation of the uncertainty it is also possible to map the transparency of the particle to the accumulated deviation: the more the particle deviates, the more transparent it will be rendered. This allows a reduction of the visibility of the particles that are either older or uncertain.

The cumulative deviation is mapped to the zero-to-one range by dividing it by v_{enc}, i.e., the highest measurable speed. Note that the cumulative deviation is unbounded, so this is a parameter that needs to be tuned. However, this mapping has shown to be robust in our experiments. Therefore, we use v_{enc} as the upper bound for the cumulative deviation, higher amounts of deviation are shown equal to a deviation of v_{enc}.

Figure 4.8 shows a streak visualization of systolic flow in a patient with aortic dissection, a vortex can be seen in the aortic arch at phases 10 to 14. The top row shows the measured flow without taking uncertainty into account, while the bottom row shows flow with our uncertainty visualization applied. Here, the cumulative deviation is encoded by an increase of the luminance while the color hue represents the seeding position on the seeding line. Note that the uncertainty is highest during diastole.

4.4. Flow exploration

The main goal of the framework is to provide tools for gaining insight in the measured PC-MRI flow, for example, its temporal behaviour and the influence of uncertainty on the visualization. The massive generation of particles can produce clutter and occlusion specially when uncertainty is considered, and as such interaction is important. In this Section, two interaction techniques are described for filtering and selection that facilitate the exploration and understanding of the flow and its uncertainty.

4.4.1. Particle transfer function

We provide a filtering mechanism to alleviate the clutter and occlusion generated by the massive amount of particles visualized. The main idea is to use the concept of transfer functions commonly used in volume rendering. We present a 2D histogram that maps the particle age to the cumulative deviation of the particles. The density in this histogram relates to the number of particles that fall within a range. Using this 2D particle-distribution histogram the distribution of the uncertainty parameters

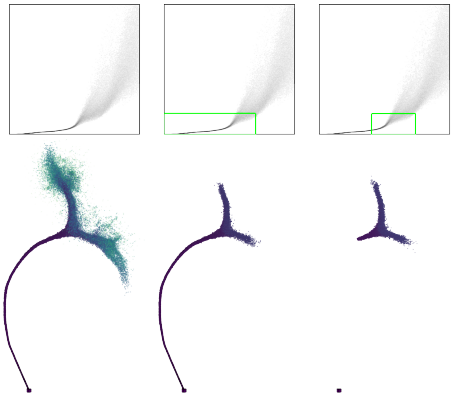


Figure 4.9: Stream visualization during peak systole. The color shows the cumulative deviation. The cumulative deviation and age of all the particles are shown in the graph on respectively the x and y-axis. Only the particles that fall in the green box area are rendered.

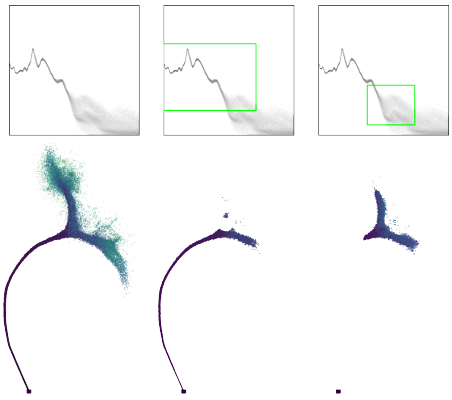


Figure 4.10: Stream visualization during peak systole. The color shows the cumulative deviation. The speed and age of all the particles are shown in the graph on respectively the x and y-axis. There is a sudden differences in particle speed that is related to a sudden increase in uncertainty. Only the particles that fall in the green box area are rendered.

give an indication of the impact of the uncertainty on the presented visualization of the flow. Furthermore it allows to filter particles that do not contribute to the understanding of the flow, for example, removing particles which are unreliable. The uncertainty is increasing monotonically over time, however, the exact form is unknown given that some phases or regions are less reliable than others. The histogram shows the form of this progression and can identify, for example, sudden increases in uncertainty, see Figure 4.9. The histogram allows, for example, selecting particles that can be considered more reliable due to long age and few deviation, or remove particles that do not contribute to the understanding of the flow. In Figure 4.9 an example is shown of the filtering. The particle-distribution histograms depicts the amount of the cumulative deviation against the particle age, which can be used to select particles of interest, based on their reliability. Similarly to a transfer function, the user can select particles in the histogram to define its opacity, and filter out all other particles, i.e., only the particles that fall into the green box of the graph are shown fully opaque, see Figure 4.9. In the future this could be extended to match the functionality of common transfer functions more. For example, optical properties, e.g. the color and opacity, could be defined based on the selection function. From this, an informed decision on the maximum line length that can be reliably assessed can be made, i.e., the line length can be based on a maximum cumulative deviation.

The particle-distribution histogram can also be used to filter particles based on their speed or other properties as shown by Figure 4.10. This can be useful to determine flow regions with particles of a certain properties. Some flow properties might work better than others depending on the goal, for example, a general assumption that is often used, is that a low velocity signal means that the data is

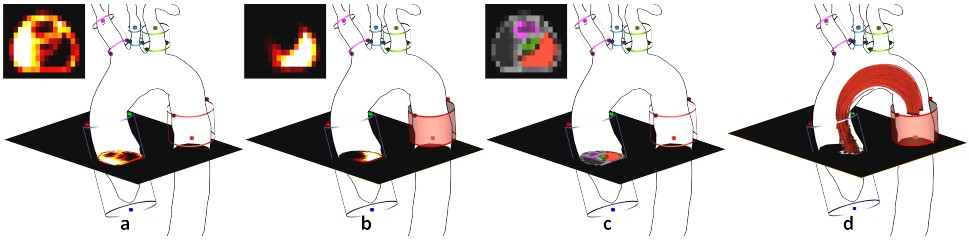


Figure 4.11: Particles were only released at the beginning of the heart cycle. Shown here is a per voxel derivation of the probability of a particle leaving the mesh (a) or through one of the branches, here the descending aorta (b). Image (c) shows the per voxel derivation of the highest probability for leaving the mesh (grey) or through the (colored) branches. Luminance of the color is mapped to the probability. Image (d) displays a pathline visualization seeded from voxels where the probability of reaching the red ring is high.

less reliable. However, some regions, such as the branches of the aorta, have a relatively low velocity signal, while the magnitude data has a relatively high signal yielding a relatively high *Signal-to-Noise Ratio* (SNR). Therefore, in this scenario, filtering based on the cumulative deviation instead of the velocity is most likely a better option as shown by Figure 4.9.

4.4.2. Particle volume distribution

In cardiac flow research, quantitative particle tracing gives information on the flow distribution, i.e., it helps to derive the percentage of blood volume that is ejected by the heart in one heart cycle and, thus, leaves the heart adequately [38]. This is achieved by ejecting particles in a region of interest and counting the amount of particles that reach other selected areas of interest. This is commonly done without considering uncertainty and, therefore, without estimation of the robustness of the result. Such an approach could also be used for aneurysms to derive the blood volume that remains within the aneurysm during a heart cycle. A similar approach for the aorta to derive the effectiveness of flow distribution can be used, e.g., to derive the probability of a particle entering one of the branches of the aorta, as was done by Friman et al. [25], in this case considering the uncertainty information. Furthermore, for finding a good seeding position it can be helpful to know where the particles flow when released at an moment in time or over a given period of time, in case of streak visualization.

In our framework, this can be computed interactively for all seeding voxels, as shown in Figure 4.11 and considering uncertainty. The user can, with minimal interaction, define an area of interest. When a particle either leaves the mesh or enters a user-defined area of interest, the reason of the particle deletion is registered by the voxel from which the particle was seeded. The total number of particles released in a given position and the number of particles that entered the user-selected area then provide an estimate of the probability of a particle seeded from this voxel to flow through the area of interest. Now, for every seeding position, the probability of a particle released from this voxel either leaving the mesh or entering a user-

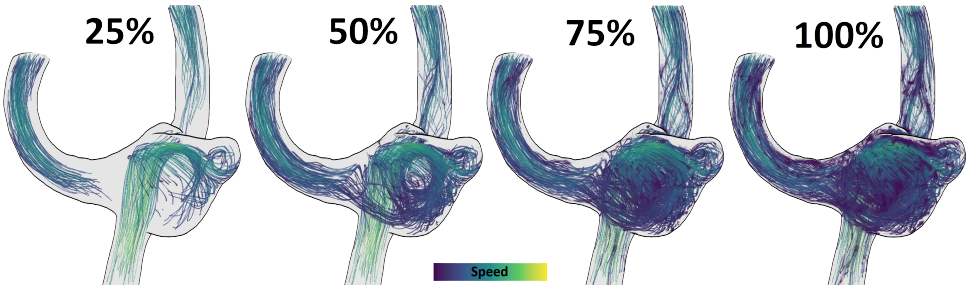


Figure 4.12: Flow within phantom data of an aneurysm. Each pathline visualization shows a different amount of lines based on the certainty of the lines. The percentage indicates how many seeding positions are used by the visualization after filtering. As such the user can visualize the most reliable flow patterns. Note that the coloring is used to encode the local speed where brighter colors represent higher speeds.

defined area, e.g., a branch, is known. This provides additional information over the usual sharp assignations of voxels to a given branch, for example, to evaluate the quality of a seeding position for a given goal. This reduces the amount of clutter since only the flow that reliably enters the region of interest is visualized, as shown in Figure 4.11. The probability of a particle leaving the mesh can help to assess the quality of the aorta segmentation, for example, a voxel from which all particles leave the mesh is most likely outside the actual aorta and should not be considered in the estimation.

4.4.3. Certainty-based seeding

Having knowledge of the uncertainty also allows for selectively visualizing only the most certain flow lines. This is done by ranking each seeding position based on the average deviation of the particles that are emerging from each seeding position. When a high number of random seeding positions is used this filtering can be used to show only the most reliable flow lines, allows that only the most certain flow patterns are visualized as shown by Figure 4.12. Moreover, by removing uncertain seeding positions, the amount of clutter caused by the uncertainty visualization can be reduced. The user can interactively change the percentage of lines shown, and thus, they can select a suiting percentage based on their interactive visual analysis.

4.5. Computational costs

The presented method provides an interactive exploration of PC-MRI data. The GPU implementation allows for advecting more than 2 million particles and rendering them at interactive rates on a system with an Intel Core i7-4770 3.4GHz CPU with 16 gigabytes of memory and an NVidia 760GTX GPU, independent of the size of measurement data and use of uncertainty information. The advection of 2 million particles for a relatively-big time step of one phase (40ms) takes less than two seconds. However, less particles are required for an ink-like visualization: typically less than 500.000 particles suffice, for which the system runs at a high frame rate.

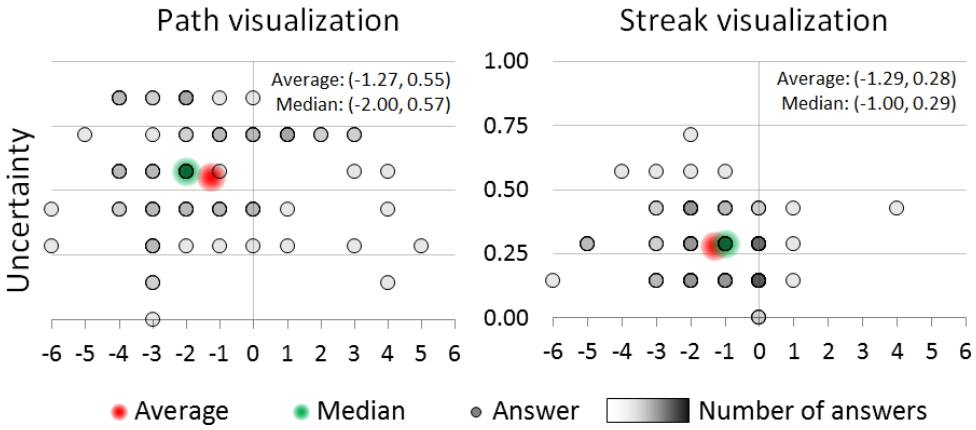


Figure 4.13: The answers of the user study for both the path and streak visualization. The y-axis represents the users uncertainty of their answer i.e. a lower value means the user is more certain, the x-axis shows the deviation from the true answer. The red and green dot represent, respectively, the average and median deviation and uncertainty. If more answers fall in the same position the point is rendered darker.

The rendering performance is linear with the number of particles, e.g., rendering 400.000 or 1 million particles takes respectively 20 and 12 frames per second.

4.6. User evaluation

Full evaluation of the variety of aspects of the presented framework is a complex task that cannot be covered within this Chapter. Therefore, we have chosen to evaluate the potential of two main aspects of our framework: the importance of using streak visualization to identify and characterize vortices, and the potential of using the framework for uncertainty visualization.

To evaluate whether streak visualization would provide additional information to the users we conducted a user study. The user study was conducted with 24 participants with various backgrounds: 18 visualization/computer graphics researchers, 4 medical students/doctors and 2 cardiovascular PC-MRI researchers. Note that, all medical students were in the last phase of their studies. A short introduction with respect to flow visualization using path and streak visualizations was given before the user study. The users were asked to count the number of vortices that formed and broke down over time in an analytic vector field based on an animations showing either pathlines or streaklines. This task is commonly used by clinical researchers to characterize the blood flow, e.g., in the heart or close to the valves [5, 131]. Note that this evaluation focuses on whether the users perceive the vortices using path and streak visualization, not whether they are able to extract all of them. For the automatic extraction and highlighting of vortices in uncertain flow data automated methods exist and could be used instead.

By using an analytic vector field the number of vortices is fixed, thus the performance of the users can be determined by the difference between their count and

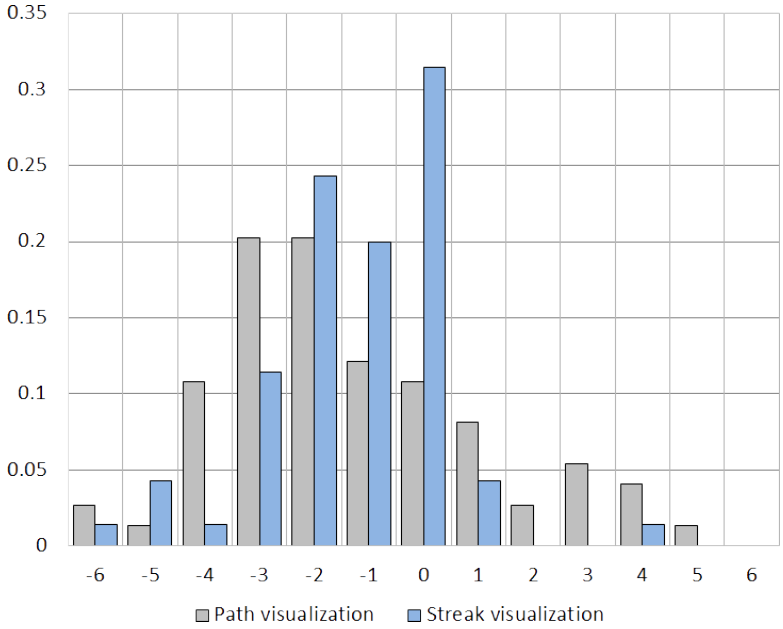


Figure 4.14: A histogram of the deviation from the actual number of vortices as counted by the users using path (grey) and streak (blue) visualization. Normalized with the total number of answers per visualization type.

the actual number of vortices. In order to obtain temporally changing flow data with a known number of vortices, a mathematically defined flow is used. A base laminar flow in a tube was created with a parabolic velocity profile, i.e., the speed increases parabolically depending on the distance to the wall of the tube. A fixed number of vortices where superimposed onto this flow throughout the spatial and temporal domain. For each vortex, the location of its core and the phase in which it forms were randomized within a range that ensured the vortex is visualized.

Since the temporal component is crucial here, only path and streak visualization were used. The streak visualization was animated over time, while the path visualization consisted of 10 stills spread out through the temporal domain. The users were presented with 6 flow visualizations in total and had to answer how many vortices they have counted (1 to 10) and how certain they were of their answer using a Likert scale from 1 to 7. To avoid the learning effect, the order, number of vortices and type of visualizations were randomized. At the end of the study the users where asked which visualization type they liked best for this task, again using a 1-to-7 Likert scale.

Figure 4.13 shows the outcome of the user study, showing both the average and median. The users where asked how uncertain they were of their answers and is mapped to the y-axis (range 0 to 1). The x-axis shows how much the answer of the participant deviated from the actual number of vortices. When asked which visualization type they like best for this task, all users showed a preference

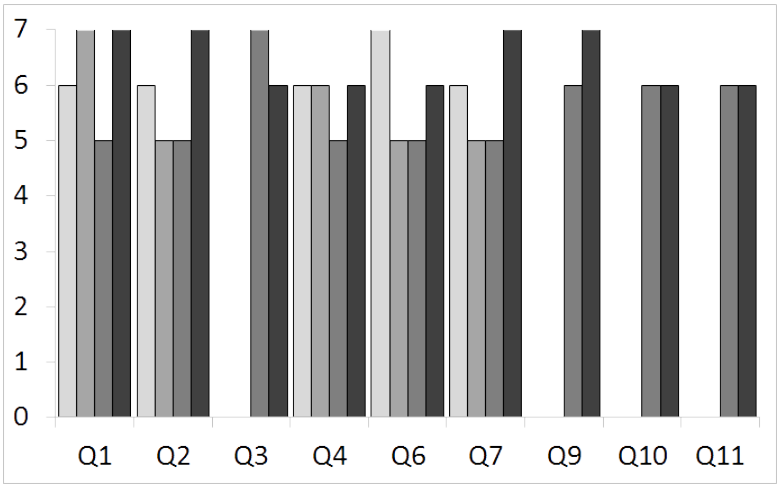


Figure 4.15: A histogram showing the agreement of the domain experts with the presented statements.

for the streak visualization with a score of six or higher out of seven. Using the streak visualization, the answers deviated less from the actual number of vortices. Furthermore, the users were generally more certain of their answers using the streak visualization. For both visualizations the users seem to rather underestimate the number of vortices.

Figure 4.14 shows a histogram of the actual deviation from the number of vortices per visualization type. It shows that the number of correct answers is higher for the streak visualization.

While path visualization is often used in flow visualization, recognition of vortices is difficult. The preliminary study suggest that the use of streak visualization can be beneficial for this task and to identify similar flow patterns, and thus justifies the use of streak visualization in blood-flow visualization. Overall, the users performed better when a streak visualization was shown. Furthermore, they were more certain that they counted the number of vortices correctly.

We also developed a preliminary user study to evaluate the potential of the uncertainty visualization and exploration provided in our framework. We presented our framework to four cardiovascular PC-MRI researchers who work with this and similar PC-MRI data on a daily basis and are very familiar with flow visualizations. We presented the clinical researchers with a questionnaire. The questionnaire was preceded by a demonstration of our visualization using synthetic data with various SNR levels. The user study was conducted in two phases. The first two domain experts were involved in determining the initial requirements for the framework. After they completed the questionnaire changes were made to improve the user study by adding additional questions. The remaining two experts were not involved in the project in any way before filling in the updated questionnaire. Both open questions and questions with respect to users' agreement to a given statement were asked. To determine the agreement a Likert scale was used where the user

could indicate their agreement within a range of 1 (negative) to 7 (positive). The questions of the user study are bellow, note that the additional questions from the second round are indicated by a black bullet instead of an open diamond.

1 ◇ Do you think visualization, in general, helps the analysis of blood flow?

2 ◇ Do you think uncertainty visualization is helpful for the analysis of blood flow?

3 • Do you understand what the uncertainty visualization represents?

4 ◇ Given the knowledge of noise in PC-MRI data, does the uncertainty visualization contribute to your confidence in your analysis?

5 • How does the uncertainty visualization influence your confidence?

6 ◇ Can you perceive the various amounts of measurement noise present in the data using the shown visualizations?

7 ◇ Would you use this uncertainty visualization?

8 ◇ For what type of analysis, if any, would you use the uncertainty visualization?

In a follow up questionnaire the use of filtering based on the uncertainty included. The users where shown examples of selectively visualizing only the most certain flow patterns before the following questionnaire:

9 • Do you think this type of filtering can be helpful?

10 • Would this type of selection influence your trust in the visualization?

11 • Do you believe this type of filtering will improve your overall analyses?

12 • How do you think the filtering could influence your analyses/conclusions?

The answers of the experts agreement to the statements are shown in Figure 4.15. Overall, the domain experts gave positive feedback towards the use of uncertainty visualization and saw a benefit in the visualization of uncertainty. Moreover, they were able to perceive the influence of the noise on the visualization and were more confident with regards to their analysis. Despite the limitations of this study, it indicates the usefulness of visualizing the uncertainty in PC-MRI data.

Based on the open question one of the users commented that the presented visualization would be specifically *interesting for the analysis of flow in an aneurysm or the aorta*. Another participant found that the visualization was useful for the *assessment of vortices (location, type, size and intensity) and wall pressure (location and force)*. Overall all participants were more confident in their analysis and trusted the visualization more. One participant mentioned that it *assists in judging the visualization, in the sense of how much one can "trust" the direction and speed shown*. Experts liked the filtering based on certainty, one mentioned that it helps them *focus on the most essential areas* and noted that *the filtering could be used to explore major flow patterns that can be classified w.r.t. to their influence on disease progression*.

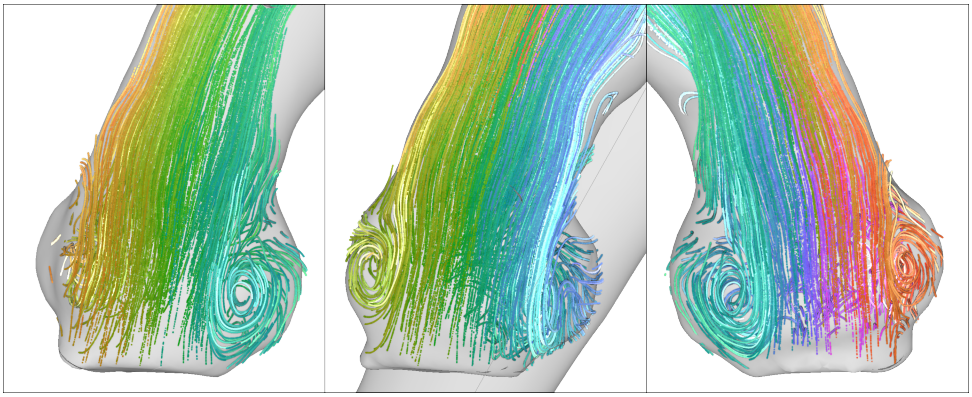


Figure 4.16: Three vortices in the aorta root were visualized with our framework. These vortices help close the three leaflets of the aortic valve.

4.7. Analysis of aorta root vortices

After being introduced to the framework, the cardiovascular PC-MRI researchers involved in the user study adopted the framework for the visualization of vortices in the sinuses or cusps in the aorta root, i.e. the *Left Coronary Cusp* (LCC), the *Non-Coronary Cusp* (NCC) and the *Right Coronary Cusp* (RCC) [19, 154]. These vortices help the leaflets open and close efficiently [5]. The goal is to detect and rate the vortices based on a semi-quantitative score. For this purpose the seeding position is determined based on the anatomical context. Since the vortices form over time, a streak visualization is the fastest option to determine at what moment the vortices are most present, if at all. The researches place a seeding disk or cylinder through the aorta root and run the streak visualization for the whole heart cycle. For scoring the vortices, a streamline visualization is used to produce a still image of the vortex in the selected phase as shown by Figure 4.16. The vortices can be color encoded per sinus based on the seeding position of the particles, while the luminance can be mapped to the particle age. Additional filtering, for example, based on the viewing direction and the centre of the seeding region can reduce clutter. This helps to distinguish the vortices and reduces confusion when interacting with the visualization in 3D. The main focus of the cardiovascular PC-MRI researchers was on post-operative blood flow in the aorta root.

As part of the study the flow was evaluated within the aortic root of patients which were operated using one of two surgical practices: the ”remodelling” procedure, as first described by Magdi Yacoub et al. [155] and the ”reimplantation” technique, as first described by Tyrone David et al. [156]. Many studies have been conducted comparing these treatment modalities and have found that mortality and post-operative aortic valve function are comparable between both surgical techniques [157].

Using PC-MRI, the vortices in the aorta sinuses after surgery can be compared to each other and healthy volunteers. Similar to before, color was used to visualize

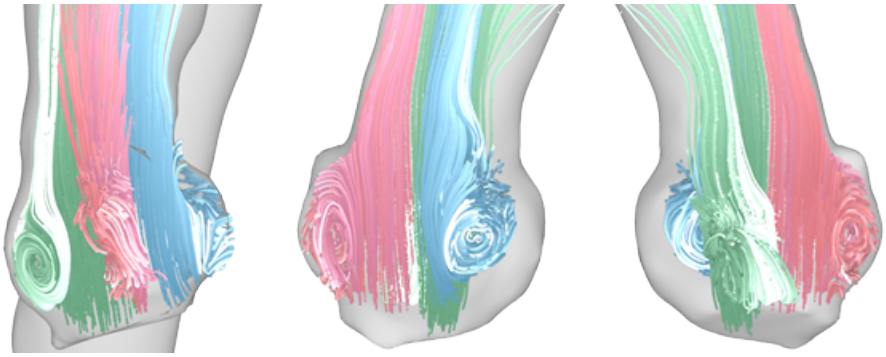


Figure 4.17: Example of the 3-dimensional visualization of each individual vortex in the sinuses of Valsalva. A three-dimensional ROI transecting the aortic valve and the three sinuses of Valsalva was defined using the 3D PC-MRA images. Each sinus vortex is depicted using streamlines in a different color based on the radial angle (RCC green, LCC pink and NCC blue).

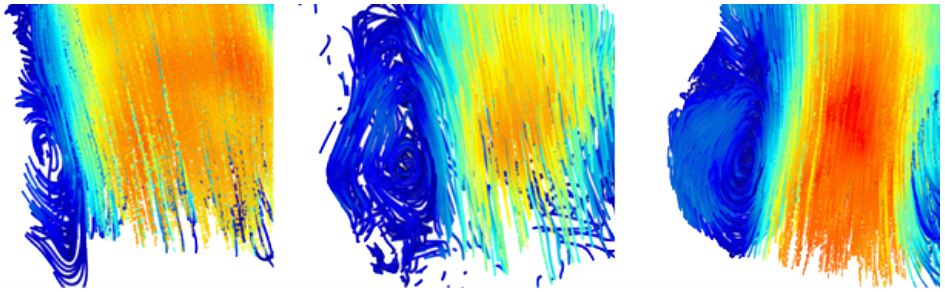


Figure 4.18: Grading scale of sinus vortices. Vortices were graded according to their size at the time point of maximum intensity (1; minimal vortex formation, 2; medium vortex formation and 3; prominent vortex formation)

the flow within each sinus, as shown in Figure 4.17.

The formation of vortices within the three sinuses of Valsalva was analysed using stream and streakline visualizations using the framework presented in this chapter. Streaklines were used to determine in which phases the vortices formed in the cusps. Because the vortices were most present for a short amount of time during a single phase streamlines were created at the various time points in which they were most clear, as shown in Figure 4.17. Vortices in all three sinuses were graded visually by two independent experienced PC-MRI observers according to their size at the time point of maximum intensity; 0 (no vortex formation), 1 (minimal vortex formation), 2 (medium vortex formation) and 3 (prominent vortex formation), as is shown in Figure 4.18.

Vortex formation

Vortex formation was assessed by our collaborators in 29 patients subjects and in each of the three individual vortices, as shown in Figure 4.19. Three individual

sinuses in three individual patients subjects could not be identified. Mean vortex scores in the entire cohort were significantly higher in the LCC and the NCC than in the RCC. No measurable differences were found between the three subgroups in overall vortex scores.

Furthermore, no differences between the various subgroups were found in each individual cusp.

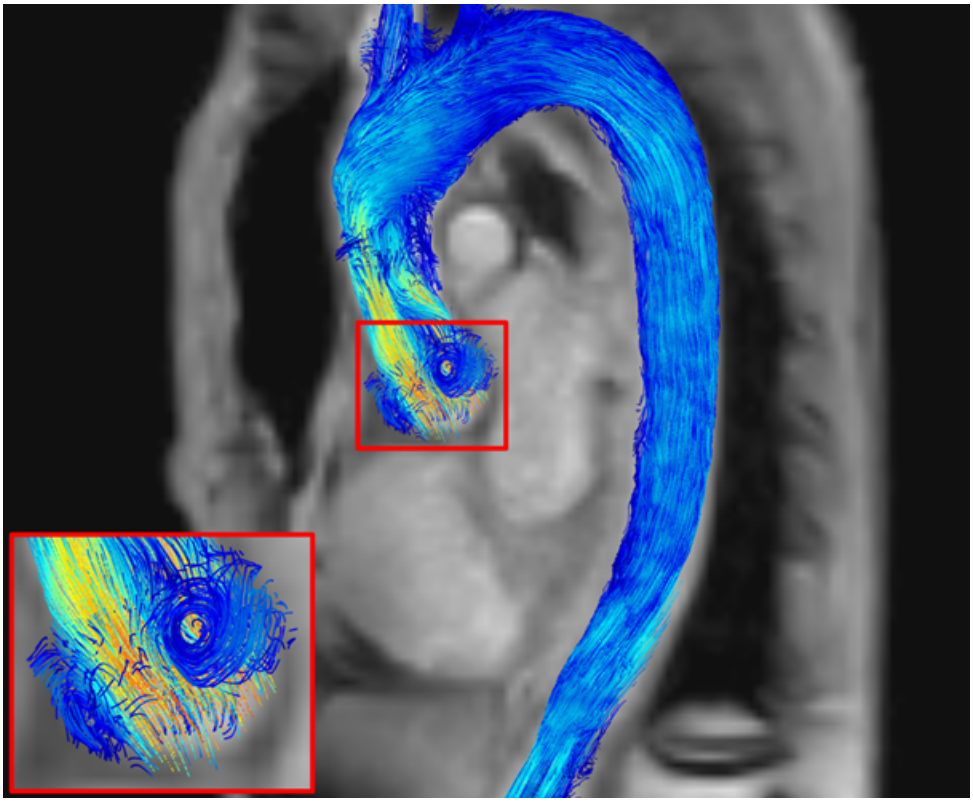


Figure 4.19: Example of the vortices in a patient after *Valve Sparing Aortic Root Replacement* (VSRR) with remodelling of the aortic root. Streamlines show prominent vortices developing in the aortic root.

Overall our collaborators found that vortex scores were significantly lower in the RCC than in the LCC and NCC. If vortices are too small, it has been hypothesized by that valve leaflets may be at risk of collision with the prosthetic aortic wall [158]. However, if vortices are too large, this may lead to systolic aortic valve dysfunction and potential leaflet remodelling [158, 159]. In this study, the effect of vortex size and vortex velocity on opening and closure patterns of valve leaflets could not be examined. These analyses are not possible using current clinical 4D flow PC-MRI techniques, due to the rapid motion of the aortic valve leaflets and the thickness of the aortic valve leaflets (less than 1 mm) in combination with the spatiotemporal resolution of current 4D flow MRI sequences [160]. However, both surgical prac-

tices, remodelling and reimplantation, are capable of preserving native-like blood flow vorticity in the sinuses. These vortices were visualized using our framework utilizing both stream- and streaklines with a visual representation specifically chosen for this use case based on the techniques explained in Section 4.2.

4.8. Summary

In this chapter a flexible framework inspired by experimental physical flow visualization was presented that allows the visualization of various aspects of blood-flow PC-MRI data. Specifically temporal behaviour and uncertainty are features that can be explored in the proposed framework and which are not commonly available in the currently used methods in PC-MRI visualization.

To mimic experimental flow visualizations, a high number of particles are traced over time through the 3D velocity fields using GPU implementation. This allows for traditional stream and path visualizations but also for streak visualizations, that can reveal time-varying flow features present in the data. Such features include vortex formation, shedding and breakdown, which are considered important factors in the development of various *Cardiovascular Diseases* (CVDs). Our collaborators adopted the framework for their study of the vortices in the sinuses in the aorta root [19]. In this study, the vortices were identified and scored based on the visualization. The flexibility of our framework allowed them to define a specific visual representation that suited this use case.

Furthermore, the framework allows for an interactive uncertainty visualization of the flow information and allows the user to visually explore the uncertainty that is present in the data. To mitigate clutter and occlusion, that is often present in 3D flow visualization, we implemented various strategies in the framework, for example a glyph-like visualization. We also provided interaction techniques such as the particle transfer functions to facilitate filtering and generation of robust visualizations. An initial evaluation among domain experts revealed that the framework is a positive addition to their analysis of the flow and they have adopted the framework to visualize the vortices that are present in the sinuses of the aorta root. Moreover, in an initial user study, they found that the uncertainty visualization contributed to their confidence in their analysis and that they were able to perceive various amounts of measurement noise present in PC-MRI data sets.

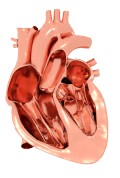
4.9. Future work

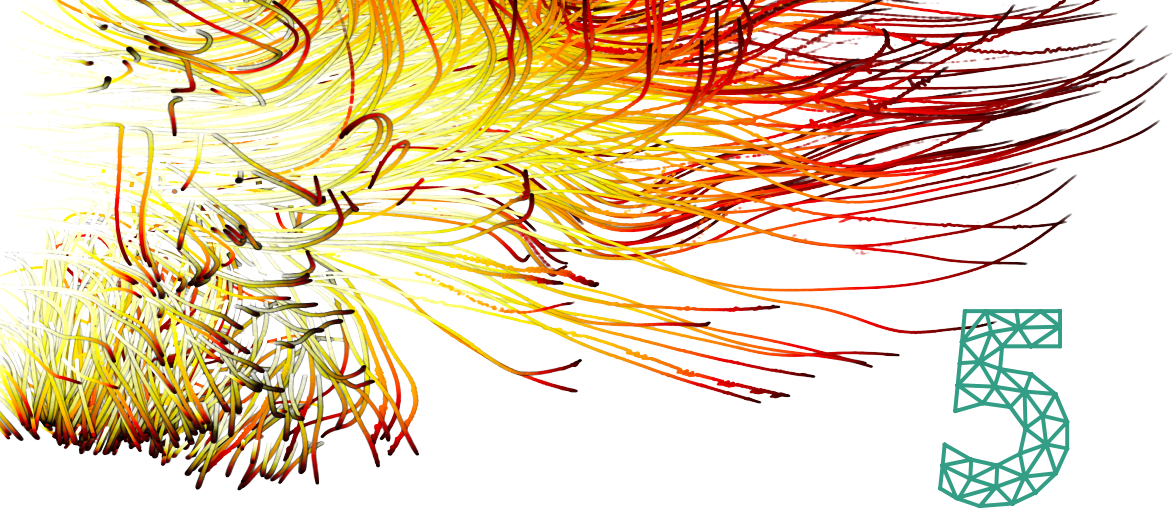
Although we believe the method improves the understanding of flow patterns and the exploration of blood-flow data, there are several open points which are subject to future work. To assess the effectiveness of the visualization, a more in-depth user evaluation, including more users providing input on how they perceive the presented visualizations would be beneficial. This is a challenging task that has been considered outside the scope of this dissertation but a promising part on the future work. For example, the evaluation of the effectiveness of the glyph-like representation and depth encoding were not considered in the evaluation. These

techniques were evaluated in a different setting in respectively Everts et al. [150] and Luft et al. [149]. However, it would be interesting to evaluate their effectiveness in the framework.

Since the uncertainty information has not been available to the domain experts before, more evaluation is needed to determine the added value of uncertainty visualization. For example, the particle volume distribution requires further clinical experiments to determine the expected amounts of flow into the aortic branches and its clinical relevance. Moreover, it would be interesting to explore how the uncertainty visualization influences the results of flow feature-based visualizations, for example, of vortices. These would require a large study, and is out of the scope of this dissertation. Furthermore, we focused on the uncertainty due to noise in the measured flow data, while other sources of uncertainty are known to influence the data, such as motion artefacts that occur in the vicinity of the moving cardiac and vessel walls. Ideally, the different sources of uncertainty should be modelled and visualized separately to identify the separate influence of each source. For example, the uncertainty resulting from the numerical integration [50, 82, 83]. Another application that would benefit from using uncertainty is the clustering of blood flow patterns that currently do not account for the uncertainty [59, 60]. By including the uncertainty, the clustering could possibly apply probabilistic similarity measures. Furthermore, it would be interesting to study the modelling of the uncertainty after pre-processing of the measured data using divergence-free filters, and how to ensure and preserve the divergence-free property within our uncertainty visualization framework. Complete modelling of sources of uncertainty is still a complex, open problem. However, all experts participating in the user study indicated that they were more confident in their analysis when the uncertainty was shown and would use the uncertainty visualization.

While the presented visualization framework can be used to visualize the complex flow patterns present in the heart, a specialized framework can be beneficial. Such a specialized visualization framework is presented in the next Chapter.





5

VISUALIZATION OF BLOOD FLOW IN THE HEART

Cardiac flow is still not fully understood, and is currently an active research topic. For the inspection of such flow, using for example the framework presented in Chapter 4, researchers often rely on methods that require additional scans produced by different imaging modalities to provide context. This requires labour-intensive registration and often manual segmentation before any exploration of the data is performed. This work provides a framework that is designed specifically for cardiac flow visualization and allows for a quick exploration of cardiac flow without the need of additional imaging and time-consuming segmentation. To achieve this, only the 4D data from one PC-MRI scan is used. A context visualization is derived automatically from the data, and provides context for the flow. Instead of relying on segmentation to deliver an accurate context, the heart's ventricles are approximated by half-ellipsoids that can be placed with minimal user interaction. Furthermore, seeding positions for flow visualization can be placed automatically in areas of interest defined by the user and based on derived flow features. The framework enables a user to do a fast initial exploration of cardiac flow, as is demonstrated by a use case and a user study involving cardiac blood flow researchers.

This chapter is based on the following publication:
A Framework for Fast Initial Exploration of PC-MRI Cardiac Flow by A.J.M. Broos, N.H.L.C. de Hoon, P.J.H. de Koning, R.J. van der Geest, A. Vilanova and A.C. Jalba [161], in *Eurographics Workshop on Visual Computing for Biology and Medicine* © 2016 Eurographics.

For the inspection of the flow, being able to select areas of interest is important to avoid occlusion and clutter. Furthermore, certain flow patterns, such as vortices, are of great interest to researchers of cardiac blood flow [162]. However, without a targeted selection of interesting areas to explore, targeted placement of seed points for the flow visualization, vortices or other flow phenomena are occluded and can be easily overlooked. Hence contextual information is important to determine regions of interest. To this end a 2D slice of the 3D morphology is often shown to provide context for the flow visualization given by the measurements, an example is shown by Figure 4.19. While 3D context visualizations for blood flow exist, they are mainly focused on vessels, for an example see Figure 4.12. These vessels do not change their shape much, and flow within them often has a clear general direction. The heart on the other hand, has a more dynamic shape over time, furthermore, the flow is highly turbulent. Therefore, a segmentation of the heart provides anatomical context, and allows visualization of the flow from and to a segments also enabling quantitative inspection. Creating such a segmentation, however, is often done manually and is a tedious and time-consuming process, making a general visualization framework such as presented in Chapter 4 less suitable. While some insight might be gained into the morphology of the heart by using the signal magnitude or the temporal magnitude intensity projection of the *Phase-Contrast Magnetic Resonance Imaging* (PC-MRI) measurements, other types of imaging techniques can provide a much higher resolution and contrast. However, registration is needed to align the data, since the scans are made at different points in time and possibly by different scanners. Furthermore, for a fast and initial exploration of the data, an initial visualization that does not require tedious segmentation would be beneficial.

Unlike the general visualization framework presented in the previous Chapter, here a dedicated framework is presented for the exploration of 4D cardiac flow PC-MRI data. This dedicated framework provides an interaction and visualization strategy to obtain a complete visualization of the cardiac blood flow based on a single 4D PC-MRI data set without resorting to time-consuming preprocessing or manual labor.

In summary, the contributions presented in this chapter are:

- Investigation and proposal of context visualization without the need of additional imaging or tedious segmentation.
- An intuitive interactive selection of the areas of interest.
- A feature-based seeding for cardiac flow visualization.
- A user study to evaluate the different aspects of the framework

5.1. Context visualization

In this section, we introduce an anatomical context visualization for the heart. The aim is to provide insight into the anatomical relation of the flow, without being too distracting. Two options are available for the basis of the context visualization:

the so-called *Temporal Maximum Intensity Projection* (TMIP) or the data of the *Phase-Contrast Angiography Magnitude* (PCA-M) of the PC-MRI measurement. The temporal maximum intensity projection of the flow speed (TMIP) contains for each voxel the maximum speed it reaches over time.

Existing straightforward 3D visualizations, such as surface or volume rendering do not work well when using the magnitude data of the PC-MRI measurement as a basis for anatomical context due to the challenging data, as shown in Figure 5.1 left. Note that both the resolution and contrast are low, resulting in less then ideal visualization of the anatomy.

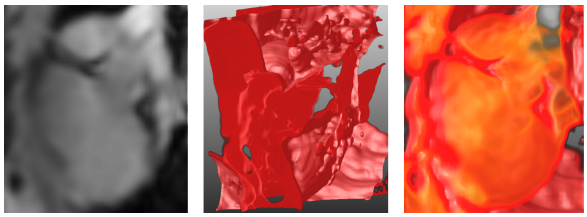


Figure 5.1: Context visualizations using the magnitude data of the PC-MRI measurement. From left to right: a single slice, an iso-surface and a volume rendering.

Often, slices of anatomical data are used as context for flow visualization. However, the heart is not two-dimensional. It takes experience and time to comprehend how such a slice corresponds to the three-dimensional anatomical data.

Another approach is to define an iso-surface based on the TMIP, it can then be used to render an iso-surface using, for example, marching cubes [39]. Using clipping techniques, for example only showing the back faces, or drawing the flow on top of the rendering, this approach can provide a good context when inspecting vessels. However, by using the TMIP the anatomy changes over time are lost, which for the dynamic morphology of the heart is less suitable compared to the more static vessels. Furthermore, the flow speed is relatively low in the ventricles and atria, resulting in low contrast for the TMIP dataset.

Illustrative techniques provide means to emphasize shape and take the focus away from unnecessary details. In this work, volume rendering is taken as a base and several techniques are implemented to improve the anatomical context visualization. Slice visualizations such as multiplanar reformat (MPR) and view-orthogonal plane visualizations are available to provide a higher level of detail.

Our goal is to obtain as much high quality context as possible given the PC-MRI data.

5.1.1. Illustrative transparency

The context visualization should not occlude the flow visualization, therefore, the volume rendering should not be fully opaque. When uniformly decreasing the opacity of the context visualization, it can become hard to perceive the morphology. Instead, illustrative transparency [62] is implemented as it emphases the contours, making the rest of the context more transparent. Contours are dependent on the view, that is, they represent the edges of structures from our point of view. See Figure 5.2 for an example of illustrative transparency, note that for illustration purposes we are using the TMIP and therefore just the main vessels are visible.

The illustrative transparency is computed for a given scalar field I. The gradient

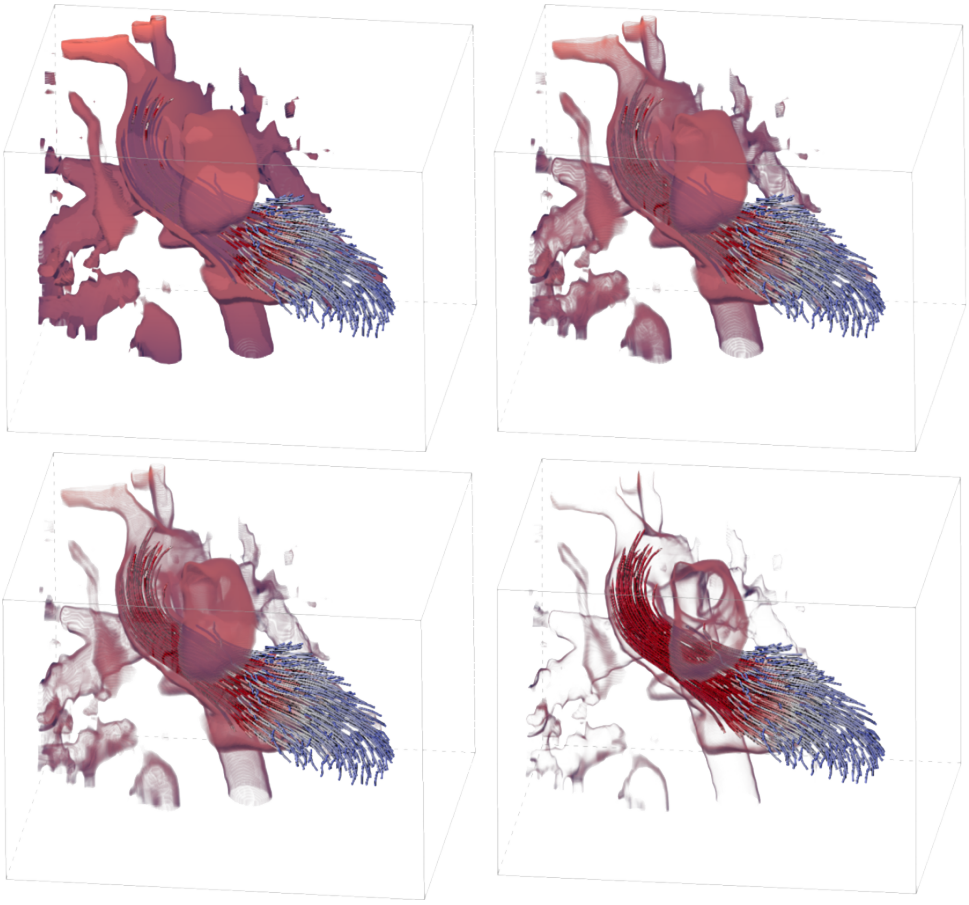


Figure 5.2: Flow from the left ventricle into aorta during systole. Context visualization using the TMIP dataset with illustrative transparency increasing from top-left to bottom-right.

vector field $\vec{G} = \nabla I$ is calculated, allowing us to look up the gradient $\vec{g}$ for a position within the field. We define a surface to be on the contour when it faces a direction somewhat perpendicular to our view direction $\vec{v}$. The direction a sample faces, i.e. the normal of a sample, is determined by the gradient $\vec{g}$ at that point: $\vec{n} = \vec{g}/||\vec{g}||$. Then, the contour variable c is defined as

$$c = \max(0, \vec{n} \cdot \vec{v} - k), \tag{5.1}$$

where k is a limit for the angle between $\vec{n}$ and $\vec{v}$, based on the curvature at the sample point. When $c = 0$ for a sample, that sample is on the contour, otherwise c has a positive value.

Contours are regulated based on the normal curvature along the view direction [61, 62]. The normal curvature along the view direction was approximated by taking the angle between the normals of two subsequent sampling points along a

sampling ray. Along the sampling ray $\vec{v}$, two points p_0 and p_1 are sampled subsequently that have normals $\vec{n}_0$ and $\vec{n}_1$. The angle α between those normals is used to approximate the curvature at p_1. We can now define the limit k as done by Kindlmann et al. [61] from Equation 5.1 using:

$$a = \frac{\cos(\alpha)}{s}, \tag{5.2}$$

$$k = \sqrt{w \cdot a \cdot (2 - w \cdot a)}, \tag{5.3}$$

where a is the approximation of the normal curvature, with s being the distance between p_0 and p_1. Then, k is the limit for contours defined by Bruckner et al. [62], where w is a parameter to control the width of the contours.

As described earlier, normals are given by the gradient of input data I. But in homogeneous regions, neighbouring gradients can have very strong variations in direction, while their magnitudes are all very low. To combat the effect of the varying normals in homogeneous regions on c, a measure h for homogeneity is defined as:

$$h = 1 - \frac{||\vec{g}||}{||g_{max}||} \tag{5.4}$$

where g_{max} is the highest magnitude gradient found in the dataset.

Now, the transparency at a sample point p_s is defined such that it has a higher value for samples further away from the contour, i.e.,

$$p_s = p_{tf}^{0.5+c \cdot h}, \tag{5.5}$$

where p_{tf} is the sample's transparency value determined by the transfer function. The constant exponent of 0.5 was determined empirically [62]. Now by increasing the transparency in the transfer function for our context visualization, we can emphasize contours and therefore the morphology's structure, as shown in Figure 5.2. Note that only vessels can be visualized, moreover, the low resolution and contrast of the magnitude data makes it less usable there.

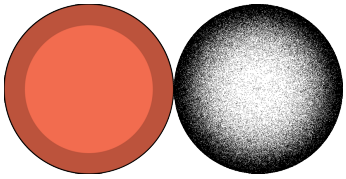


Figure 5.3: Examples of *Lit Sphere Maps* (sLSMs). The pink LSM (left) was drawn by hand and the dotted one (right) was generated.

LSMs [68] are textures that define a lighting and coloring style on a sphere, as shown in Figure 5.3. Providing an easy and effective way to generate illustrative renderings. The normal of every sample is projected onto a plane orthogonal to the view direction. When placing the centre of the LSM at the base of the normal vector, the end of the vector projects onto a pixel in the LSM. This pixel determines the color of the sample. This enables different illustrative styles for the context visualization. In Figure 5.4, different LSMs are applied. Due to their abstract nature, the LSMs provide a context visualization without distracting the user from the flow.

5.1.2. Lit sphere maps

We propose to generate a dotted lit sphere map as follows. A uniform distributed random sample $u \in [0, 2\pi]$ is taken. Another sample w is taken from an exponential

distribution with probability density function $f(a) = \lambda e^{-\lambda a}$, where $a \in [0, \infty)$. We then calculate positions of black pixels $(x, y) \in [-1, 1] \times [-1, 1]$ such that the angle is uniformly distributed and the radius exponentially:

$$r = 1 - w \tag{5.6}$$
$$x = r \cdot \cos(u) \tag{5.7}$$
$$y = r \cdot \sin(u) \tag{5.8}$$

Note that if w exceeds 1, another sample is drawn. This process is repeated N times, with N being the number of pixels in the image. Using the above technique ensures more black pixels will be at the border of the disk, resulting in an emphasis of the contour. For the dotted image in Figures 5.3 and 5.4 the image size is 512×512 and $\lambda = 8$.

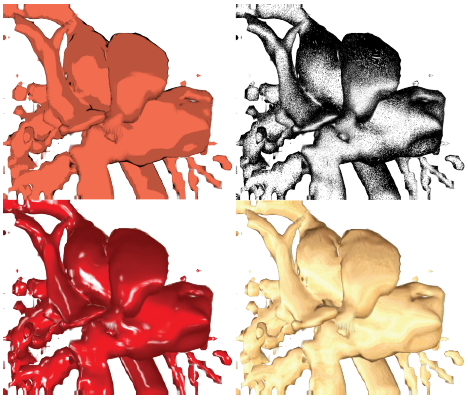


Figure 5.4: Context rendered with different LSMs. Top row, from left to right: flat pink, dotted, bottom row, from left to right: muscle and bone. The muscle and bone-like LSMs are based on Bruckner et al. [62].

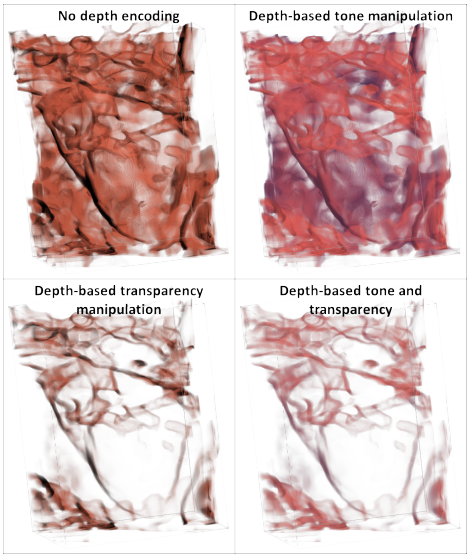


Figure 5.5: Comparison of context visualization using different depth-based manipulations.

5.1.3. Depth enhancement

The above illustrative volume rendering techniques do not encode the depth, making it difficult to discern spatial relations. For this purpose, two techniques were implemented, depth-based tone manipulation and depth-based transparency manipulation.

Depth-based tone manipulation uses color to give samples at the start of the ray a warmer color, while giving samples at the end of the ray a cooler color. The depth cue comes from the perception that cool colors recede while warm colors advance [163, 164]. Here, a tone from yellow (warm) to blue (cool) is used to map from near to far, respectively.

Depth-based transparency changes transparency based on depth. We found that the transparency at a sample point p_t defined by $p_t = p_{tf} \cdot \sqrt{d}$ works well. Here, $d \in [0, 1]$ is the depth, and p_{tf} is the sample's transparency value determined by the transfer function.

Combining both depth encodings does not bring much improvement since the transparency manipulation reduces the effect of the cooler colors in the back. Figure 5.5 shows the two methods used in combination with a a volume rendering of the heart.

We have generated several context visualizations based on TMIP or PC-MRI magnitude data to investigate the use of this information as context.

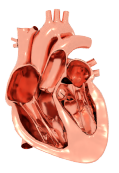
5.2. Ventricle approximation

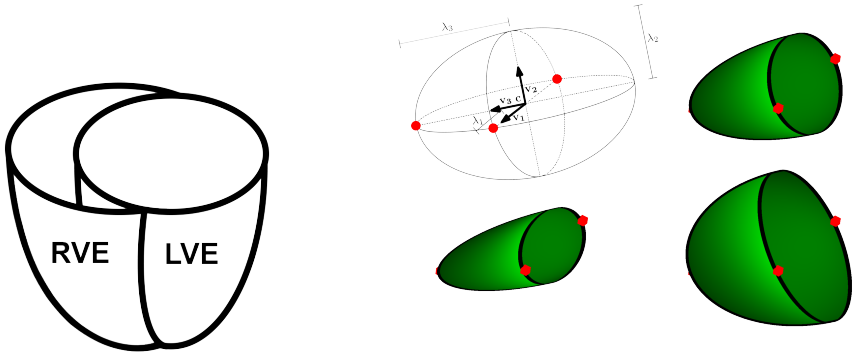
Seed points are often needed for flow visualization techniques, e.g., pathlines and particles. Placing such points manually is tedious and time-consuming. Therefore, seeding regions are commonly defined wherein seed points are automatically generated.

A key observation by Kovalova et al. [76] is that the ventricles can be approximated by half ellipsoids; i.e., the left ventricle has a shape similar to half of an ellipsoid, while the right ventricle is wrapped around it, as seen in Figure 5.6a. Here, *Left Ventricle Ellipsoid* (LVE) is the set of voxels inside the left ventricle ellipsoid and *Right Ventricle Ellipsoid* (RVE) are the set of voxels inside right ventricle ellipsoid. The ventricles will be modelled by ellipsoids without any account for wall thickness. Here, the set of voxels inside the right ventricle is defined by RVE $\setminus$ LVE. Notice that for the purpose of seeding and visualization an approximation of the location should in general be sufficient.

Tri-axial ellipsoids can be placed by the user; an overview is shown in Figure 5.6b, where the centre point is denoted by c, its orthonormal axes are $\vec{v}_1$, $\vec{v}_2$ and $\vec{v}_3$, and the radii along those axes are given by λ_1, λ_2 and λ_3, respectively. The half ellipsoid can be defined by the user using three guide points. For the placement of these points the user can inspect the magnitude and TMIP using the multiplanar reformat visualization or the view-orthogonal plane visualization. The user places the guiding points g_1, g_2 and g_3 on these planes which then define a half ellipsoid. One point represents the apex and the other two points should be near the atrium of the corresponding ventricle.

The centre point c is derived as $c = (g_1 + g_2)/2$. Furthermore, $\vec{v}_2$ is defined as the vector pointing towards the view and $\vec{v}_3 = c - g_3$. $\vec{v}_1$ is expected to be orthogonal to the other vectors, hence it is computed as $\vec{v}_1 = \vec{v}_3 \times \vec{v}_2$. Since $\vec{v}_1$ is orthogonal to $\vec{v}_3$, it is possible that the half-ellipsoid does not line up with the first two guiding points. Therefore, a clipping plane is used that passes through those two points. λ_1, λ_2 and λ_3 can be derived from the distance between the centre and guide points. Since λ_2 is not defined by the guide points it will initially be set to λ_1, however, the user can easily vary this value to squeeze or stretch the ellipsoid. This allows users to the define half ellipsoids to estimate the ventricles with little user interaction.





(a) Geometric approximation of the ventricles using ellipsoids: the left ventricle ellipsoid (LVE) and the right ventricle ellipsoid (RVE). The right ventricle is approximated by RVE \ LVE.

(b) A tri-axial ellipsoid with center point c. The orthonormal axes are given by $\vec{v}_1$, $\vec{v}_2$ and $\vec{v}_3$. The radii along those axes are λ_1, λ_2 and λ_3. Initially $\lambda_2 = \lambda_1$ (top right). The user can vary λ_2 to squeeze (lower left) or stretch the ellipsoid (lower right).

Figure 5.6: A simple approximation of the two ventricles using ellipsoids.

The seed points can now be generated within LVE and RVE \ LVE to investigate the flow in both ventricles. To avoid occlusion, only the contours of the ellipsoids are rendered throughout this paper.

5.3. Feature-based flow visualization

When inspecting cardiac flow, researchers are often interested in certain flow features, such as speed or vorticity. Our feature-based seeding approach aims to facilitate quick identification of such areas, and automatic placement of seed points similar to Stalling et al [56].

Using transfer functions the user can directly influence the seeding probability of the location where a value of the selected flow feature occurs. That is, for every voxel v, the value v_f for a certain feature f can be calculated. Based on that value, a seeding probability $p_{v_f} \in [0, 1]$ is determined by the transfer function. Multiple transfer functions can also be combined.

The following features are provided:

- Flow speed magnitude
- Magnitude data (PCA-M)
- Curl
- λ_2-criterion
- Q-criterion

Measures of vorticity are often used to inspect cardiac flow. Such rotation of flow can be computed using the curl, which describes the rotation of a 3-dimensional vector field. The curl of a vector field is a vector field itself. Every vector represents the axis of rotation, according to the right-hand rule, and the magnitude of the curl vector represents the amount of rotation.

To compute the curl we compute the gradient tensor of the vector field F at position $\vec{x}$:

$$G(\vec{x}) = \begin{bmatrix} \frac{\partial F_x}{\partial x} & \frac{\partial F_x}{\partial y} & \frac{\partial F_x}{\partial z} \\ \frac{\partial F_y}{\partial x} & \frac{\partial F_y}{\partial y} & \frac{\partial F_y}{\partial z} \\ \frac{\partial F_z}{\partial x} & \frac{\partial F_z}{\partial y} & \frac{\partial F_z}{\partial z} \end{bmatrix} \tag{5.9}$$

From this, the curl at position $\vec{x}$ is given by:

$$\vec{curl}(\vec{x}) = \begin{bmatrix} \frac{\partial F_z}{\partial y} - \frac{\partial F_y}{\partial z} \\ \frac{\partial F_x}{\partial z} - \frac{\partial F_z}{\partial x} \\ \frac{\partial F_y}{\partial x} - \frac{\partial F_x}{\partial y} \end{bmatrix} \tag{5.10}$$

Often, the core of the vortex, the region around which the vortex rotates, is of interest. Such areas are often found using both the so-called λ_2-criterion and the Q-criterion. For the computation of the λ_2-criterion the rate-of-strain tensor S and vorticity tensor Ω are computed:

$$S(\vec{x}) = \frac{G(\vec{x}) + G^T(\vec{x})}{2} \tag{5.11}$$

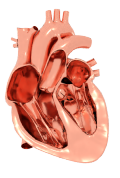
$$\Omega(\vec{x}) = \frac{G(\vec{x}) - G^T(\vec{x})}{2} \tag{5.12}$$

From these, the three eigenvalues $\lambda_1 \geq \lambda_2 \geq \lambda_3$ of $S(\vec{x})^2 + \Omega(\vec{x})^2$ are computed. If two eigenvalues are negative for a given point, that is $\lambda_2 < 0$, it lays in a vortex core. Similarly, for the Q-criterion if the value of Q is positive the point is part of a vortex core, where Q is given by

$$Q(\vec{x}) = \frac{||S(\vec{x})||^2 - ||\Omega(\vec{x})||^2}{2} \tag{5.13}$$

The user can use a transfer function to define the probability of a seed being placed in a voxel based on its value for the selected feature. An example of feature-based seeding is shown in Figure 5.7. Here the region shown in magenta depicts the area where $\lambda_2 < -30$. This region is used to distribute seed points for pathlines. A combination of features can be used by defining multiple transfer functions for multiple features. The probabilities are in this case multiplied to obtain the seeding probability per voxel.

5.4. Evaluation



Our application was evaluated using a use case and a user study held among 6 4D PC-MRI flow researchers.

For the evaluation, two datasets were used: a healthy volunteer and patient data. For the healthy volunteer 30 phases were measured covering a single heartbeat, yielding a temporal resolution of 32ms. Providing $65 \times 55 \times 25$ voxels with a size of $2.27 \times 2.27 \times 4.20$mm. The patient data covered a full heart cycle in 30 phases with a temporal resolution of 40ms. Resulting in 30 volumes with $70 \times 55 \times 38$ voxels each sized $2.27 \times 2.27 \times 3.00$mm.

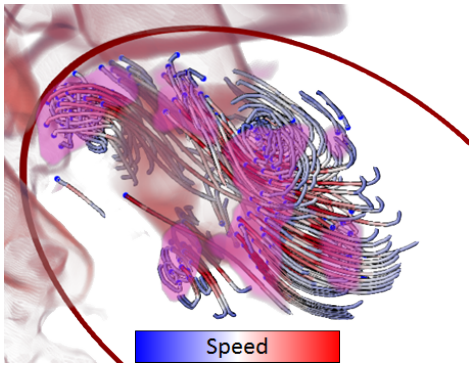
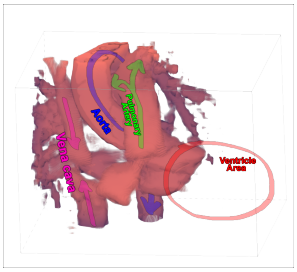


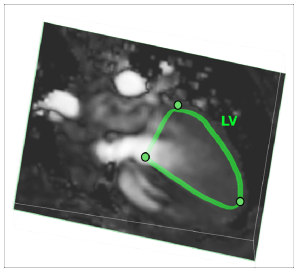
Figure 5.7: Pathlines seeded (blue points) using the λ_2-criterion, showing several vortices inside the left ventricle during early diastole. The area shown using magenta is the seeding region, where $\lambda_2 < -30$.

5.4.1. Use case

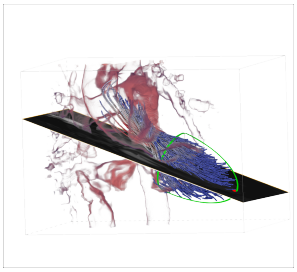
For this use case, the context visualization is used to obtain a reference with the vessels. The context visualization, however, does not clearly show the ventricles themselves, as shown in Figure 5.8a. Therefore, a slice depicting the speed is aligned with the vessels, shown in Figure 5.8b. The selected slice is of a phase during systole, so the aorta shows in bright white; this eases the search for the left ventricle (denoted by LV). The view-orthogonal plane is used to place the slice approximately through the middle of the LV. Guide points are placed on the slice to create the LVE in Figure 5.8c. The opacity on the context visualization has been decreased, to allow us to see flow within the vessels, while the contours are still emphasized. After hiding the slice, the flow is inspected further. During early systole, we notice that not all the blood seems to flow into the aorta. Some of the flow seems to go back into the direction of the left atrium, as shown in Figure 5.8d. To further investigate, another ellipsoid is placed. By tracing the pathlines backwards through time from the left ventricle, the left atrium can be found at a phase in diastole, see Figure 5.8e. The second ellipsoid can now be placed on the slice. Using forward tracing pathlines, the flow into left ventricle can be seen in Figure 5.8f suggesting correct placement of the ellipsoid. Since the position of the unexpected flow in Figure 5.8d and Figure 5.8f match, this indicates the need for further inspection of the so-called mitral valve. This finding was confirmed by the researchers at (LUMC). This is further supported by Figure 5.8h which shows flow behind the mitral valve back into the left atrium, while in a healthy volunteer, shown in Figure 5.8g, the blood behind the mitral valve remains relatively still. If not all blood flows towards the aorta during systole, but it flows in the left atrium this indicates a dysfunction of the mitral valve.



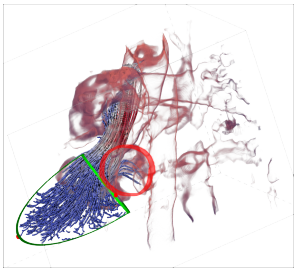
(a) Our context visualization showing the vessels. Here magenta shows the vena cava, blue the aorta, green the pulmonary artery and red indicates the expected location of the ventricles.



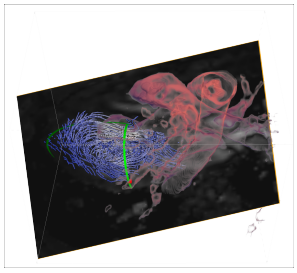
(b) A slice at systole through the vessels. On the slice white means high flow speed while black represents low speeds. The green outline denotes an approximation of the Left Ventricle.



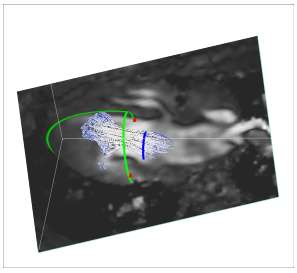
(c) Placement of the ellipsoid in the left ventricle. The pathlines show the blood flow from the LVE into the aorta at peak systole.



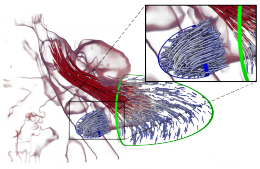
(d) Further expection shows blood flowing both in the aorta and the left atrium as indicated by the red circle. This suggests an aberrant flow pattern.



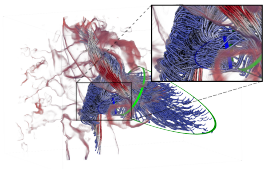
(e) Backward tracing pathlines from the left ventricle helps the alignment of a slice with the incoming flow.



(f) Placement of a second ellipsoid, corresponding with the mitral valve. Flow is shown with forward tracing pathlines.



(g) Flow during peak systole in a healthy volunteer. The mitral valve prevents the flow from the left ventricle into the left atrium.



(h) Flow during peak systole in a patient. Blood flows to both the left atrium and left ventricle.

Figure 5.8: Steps followed in the use case. The coloring of the pathlines indicates speed: from blue (slow) to white to red (fast).

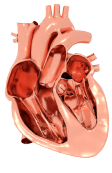


Table 5.1: Statements on the framework presented to 6 domain experts and their agreement with the statements.

Statement	R1	R2	R3	R4	R5	R6	Agreement
The general structure of the heart can be perceived from the context	4	4	4	5	4	4	+
The context visualization helps with understanding the flow	2	4	5	4	5	4	+/-
The slices complement the volume rendering	3	5	4	4	5	4	+
The left ventricle can be approximated by LVE	5	4	4	4	4	4	+
The right ventricle can be approximated by RVE \ LVE	4	4	4	3	3	4	+
The placement and interaction with the ventricle ellipsoids is intuitive	5	4	4	4	4	4	+
Using an ellipsoid for clipping (compared to clipping planes) is suitable for heart data	4	4	4	4	3	4	+
Automatic seeding in areas of interest can be done based on flow features	5	5	4	4	4	4	+
Transfer functions are an intuitive way specifying the properties of areas of interest	3	4	3	4	4	4	+/-
The project enables exploration, providing an overview of cardiac flow before using other methods for closer inspection	4	5	4	5	4	4	+

5.4.2. User study

The goal of this framework is to enable initial cardiac flow exploration without time-consuming manual preprocessing. To see whether the visualization generated would fulfil the exploratory needs, 4D flow experts were questioned. Six researchers from the cardiology and radiology departments of the LUMC filled out a questionnaire. Before the questionnaire, they were given an overview of the project by means of a small presentation. Then, a demonstration per single component of the tool was shown, after which the experts filled in the questionnaire for that specific component. The project was divided into context visualization, ventricle ellipsoids, and feature-based seeding components. Finally, they were asked for their general opinions and suggestions. The questionnaire can be found in the additional materials. In this section, the results of the questionnaire are presented. A Likert-scale was used in the questions that were formulated as a statement and the researchers could tick a box depending on their agreement with the statement. The results of the questionnaire are shown Table 5.1. Here, + indicates that many of them agreed, +/- indicates that opinions differed, while - indicates that they disagreed with the statement.

Questions involving choices between options presented in figures and open questions will be discussed separately.

5.4.3. Context visualization

Two options were available for the basis of the context visualization: the TMIP or the PCA-M datasets. There was a unanimous preference for the TMIP data as a basis. All the researchers agreed that the general structure of the heart could be perceived from the context, whereas all except one agreed that it helps with understanding the flow. The ventricles and atria were not visible in the context visualization, however, all agreed that the slices provide this additional information. Especially the view-orthogonal plane was much appreciated.

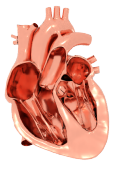
While depth-based transparency manipulation helps with the PCA-M based context visualization, depth-based color manipulation was preferred for depth perception by all researchers. Several options for the lit sphere maps were presented, with the favourite being the flat pink LSM with four votes, shown top left in Figure 5.4. The remaining two votes went to the bone-like LSM, also shown in Figure 5.4 (bottom right). When asked if they would use the context visualization, 4 said yes. One was not sure, and the remaining expert thought it was not applicable to their work.

5.4.4. Ventricle approximation

All researchers agreed that in general the ventricles could be well approximated using the ventricle ellipsoids. However, for rare patients with, for example, a single ventricle defect it might not provide a good approximation. Placing three guiding points on a slice seemed intuitive to the researchers. They also agreed that a clipping ellipsoid can be a better fit for the shape of a heart, compared to planes or boxes. When asked if they would use the ventricle ellipsoids, all agreed that they would use it. Even for the less appropriate cases; some patients have severely different morphology for the ventricles, they might still work well for defining the region of interest.

5.4.5. Feature-based flow visualization

All 6 researches agreed that areas of interest could be defined using flow features and seed points could be placed within them automatically. While most agreed that transfer functions were an intuitive way to specify properties of these areas of interest, two were neutral on this part. Clinical use was mentioned, however, would be limited since doctors would not be familiar with transfer functions. Opinions were divided when asked whether they would use feature-based seeding. While areas of interest could be specified, some would not want to automatically fill the whole areas, but rather use it as a guide for manual placement of seed points. Opinions also differed about the need for more than two features being combined. Other features that were suggested to be interesting are ones related to wall-shear stress, velocity direction compared to a given vector, and helical flow.



5.4.6. General feedback

The goal of this work was to enable exploration of cardiac flow for a quick initial overview. It was unanimously agreed that this goal was achieved. 5 experts thought there was a need for such software, the remaining expert was not sure. However, two researchers noted that the research side generally is not interested in new ways of qualitative inspection anymore. Both mentioned that the framework would be very suitable for clinical, on-site use. Suggestions for future work included improving the user interface for clinical use, more flow visualization options compared to just pathlines and particles, trying the software on many different heart diseases, and interpolating the ventricle ellipsoids over time between two placements: one for systole and one for diastole.

5.5. Conclusion and future work

The aim of this chapter was to provide a framework that aids a fast exploration of cardiac flow. For this purpose, anatomical context visualization for flow was explored. While direct volume rendering of the magnitude data was initially to be the basis of the context visualization, this data was found to be unsuitable as a context in most situations. The temporal maximum intensity projection, TMIP, led to a better visualization of the context. It provides a good initial point of orientation, and provides a clear context for flow in the main vessels around the heart, which also aids in identifying the general location of the heart. Applying illustrative transparency the structure of the vessels remains visible while obstructed view on the flow inside is minimized. Different lit sphere maps allowed for easy switching between different styles of visualization for the context. To improve depth perception, depth-based tone manipulation and depth-based transparency manipulation work well encoding the depth, improving the context visualization. The direct volume rendering of the TMIP data provides no information about the current phase of the heart cycle, nor a clear distinction of the ventricles and atria. This is evaded by a slice-based visualizations by means of a multiplanar reformat and a view-orthogonal plane, on which the magnitude and velocity data can be shown.

The ventricles can be approximated by half-ellipsoids. Our ventricle ellipsoids definition use an intuitive method of interaction and can easily be placed. While the ventricle ellipsoids, as their name suggests, are intended for approximating the ventricles, they are not limited to that purpose. This was clearly demonstrated in the use case presented in this work, where one was placed behind the mitral valve to investigate its deficiency. For the clipping of data or the visualization, usually done using clipping planes, the half-ellipsoid proved to work well, since it fits the shape of the heart and structures within it much better, further enhancing the context visualization.

To aid in identifying interesting areas for flow inspection, and to automate seed point placement within such areas, feature-based seeding was implemented. Different features determining the area of interest can be chosen. The corresponding transfer functions made for an interactive method to define the interesting ranges of values. Using feature-based seeding, one can quickly identify regions with, for

example, high speed vortices. The probability-based design allows to easily scale up to combinations of different features.

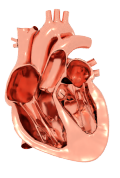
There are several points in which the framework could be extended. For example, the particle approach from Chapter 4 could be integrated, allowing for more flexibility. Moreover, the context visualization could be improved such that the ventricles and atria can be shown. If no additional imaging is desired, this would require a filtering and segmentation of the existing data. Using the half-ellipsoids helps to provide a context for the flow visualization. However, these remain static over time, while the anatomy of the heart changes, rendering the approximation less suitable. Furthermore, using the half-ellipsoids does not aid to distinguish the atria and the ventricles clearly.

The ventricle ellipsoids can be used for other purposes as well. One could trace particles placed inside the ventricle, perform both backward and forward tracing and then classify them based on the resulting positions [165]. This would provide additional information such as whether a particle enters the ventricle and leaves within one cycle or whether it stayed within the ventricle throughout the heart cycle.

The flow visualizations and seed placement provided could also be improved upon to avoid cluttering using for example techniques like hierarchical clustering of the flow [58], dashtubes [166], or a combination with image-based seeding [167].

A more thorough user study including more then 6 specialists would be beneficial. Such an user study should include tasks for the users to perform using the framework. Timing of these tasks then provides insight the learning curve of the framework.

Despite its limitations, the proposed framework allows researchers to explore the flow data without time consuming preprocessing and get a first insight on the obtained data. In the next Chapter, we take a step forward to improve the PC-MRI data itself, and therefore the visualization of the data, by combining measured PC-MRI data with simulations to remove noise and increase the temporal resolution.



6 Coupling simulation and measured PC-MRI data

In Chapters 4 and 5, the focus was on the direct visualization of PC-MRI data for different situations. However, these measurements are susceptible to artefacts, noise and a coarse spatio-temporal resolution. Furthermore, typical flow visualization techniques rely on interpolation. In this Chapter, we will focus on the accuracy and robustness of the data by introducing data assimilation methods. Hereby, a unique way of analysing patient-specific hemodynamics becomes possible. To this end, besides imaging data, also numerical simulations are employed. These are based on mathematical models of specific features of physical reality. However, these models require realistic parameters and boundary conditions based on measurements. Data assimilation is used to bring measured data and physically-based simulation together, and to harness the mutual benefits and steer a physically-based simulation of the flow. The accuracy and noise robustness of the coupled approach is validated using an analytic flow field. Furthermore, we present results that show the differences between using conventional interpolation and our coupled approach.

This chapter is based on the following publications:
4D MRI flow coupled to physics-based fluid simulation for blood-flow visualization by N.H.L.C. de Hoon, R.F.P. van Pelt, A.C. Jalba and A. Vilanova in *Computer Graphics Forum (2014)* [168] © 2014 Wiley.
Temporal Interpolation of 4D PC-MRI Blood-flow Measurements Using Bidirectional Physics-based Fluid Simulation by N.H.L.C. de Hoon, A.C. Jalba, E. Eisemann and A. Vilanova [169], in *Eurographics Workshop on Visual Computing for Biology and Medicine* © 2016 Eurographics.

Phase-Contrast Magnetic Resonance Imaging (PC-MRI) imaging data provides important patient-specific information, however, they are prone to noise and artefacts as previously explained in Chapter 2.2.3. Also, the spatial resolution remains limited, leading to partial volume effects. This causes poorly defined velocity vectors near the walls, because boundary transitions occur at sub-voxel scale. The most pressing issue for the analysis is, however, the coarse temporal resolution.

To date most blood-flow visualizations employ *linear interpolation* (lerp) to approximate values between the coarsely spaced time points. Many visualization techniques, such as line and particle traces, rely on numerical integration methods that use interpolated values. The physical knowledge about the blood-flow behaviour as explained in Chapter 2.3 is currently not employed to improve the approximation of inter-measurement velocity information.

To harness the advantages of both blood-flow measurements and simulations, we advocate for a viewpoint where both approaches are coupled, through the generic technique of data assimilation – the process of combining governing principles with potentially sparse, noisy and/or irregularly-distributed data. The concept of data assimilation is well established in fields such as geophysics and meteorology, where sparse and noisy data are combined with dynamics principles to obtain accurate predictions of physical phenomena, see e.g., Ghil and Malanotte [170]. In this chapter, a novel method is presented for improving temporal resolution of PC-MRI data for flow visualizations. The method is based on the coupling of PC-MRI measurements and physics-based fluid simulations that stems from computer graphics research. Such methods, typically used for movies and games, provide an unmatched computational performance, and are nowadays typically based on physical knowledge, modelling the incompressible fluid in a well-defined boundary, while conserving mass and energy. In particular, we use a hybrid fluid simulation, combining a grid-based and particle-based approach. We strive to improve the temporal interpolation. Moreover, important flow features, such as vorticity, must remain present.

6.1. Simulation algorithm

We use the FLIP simulation approach as introduced in Section 2.4.1. The main steps of the blood-flow simulation are shown in Algorithm 1. The simulation is initialized with the first PC-MRI measurement. The time step of the simulation is bounded by the *Courant-Friedrichs-Lewy* (CFL) condition that ensures that a particle moves at most one voxel per time step, Δt, to reduce the numerical error, i.e.,

$$\Delta t = \frac{voxelSize}{v_{enc}}. \tag{6.1}$$

Here Δt is the maximum safe time step in seconds, $voxelSize$ is the smallest dimension of the voxels in meters, 0.002 meters (2mm) in our data sets, and v_{enc} is an acquisition parameter (in m/s), representing the largest speed that can be measured unambiguously. By using the CFL condition empty cells can only occur next to non-empty ones, since the fluid cannot move more than one grid cell per time step.

Algorithm 1: Blood-flow simulation algorithm.

1 Initialize positions and velocities of *Fluid Implicit Particle* (FLIP) particles;
2 **foreach** *simulation time step* **do**
3 Compute velocity of each grid cell as a weighted average of nearby particle velocities;
4 Advect particles through the grid velocity field using a second-order Runge-Kutta *Ordinary Differential Equation* (ODE) solver; also, constrain particles to lie inside vessel walls;
5 Extend fluid-cell velocities to nearby non-fluid cells;
6 Make the grid velocity incompressible;
7 Add to particle velocities the differences between grid velocities at steps 3 and 6;
8 Create new FLIP particles;
9 **end**

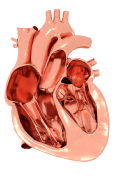
During the simulation, new FLIP particles are created in empty (non-fluid) grid cells surrounding non-empty (fluid) ones, step 8. The velocity at these cells is initialized by extending the velocity field of the fluid using the Fast Sweeping Method [171], see step 5 of Algorithm 1. The new particles get their velocities from the grid cells they occupy, by using an approach similar to *Smoothed Particle Hydrodynamics* (SPH) interpolation [172]. If a measurement is available at the current time step, the measured velocity field is used instead of the simulation field.

The density field ρ is evaluated on the grid using SPH interpolation of particles' masses. In step 6, the grid velocity is made incompressible using the variational method by Batty et al. [34]. One advantage of this method is that it avoids *locally* discretizing the sensitive boundary condition, employed by standard pressure *Partial Differential Equation* (PDE) solvers at non-grid-aligned vessel walls. Instead it solves the normal equations for pressure – a *global*, consistent, symmetric positive semi-definite system given in Equation (2.10).

6.1.1. Irregular boundaries

For the vessel walls we use $\mathbf{n}$ and $\mathbf{v}_{wall}$ being the unit normal and velocity of the solid vessel boundary, respectively; additionally we assume $\mathbf{v}_{wall} = 0$ (static walls). Both the 'no-slip condition', or a Dirichlet boundary condition and the 'no-penetration condition', or a Neumann boundary condition can be used. The no-slip condition means $\mathbf{u} = \mathbf{v}_{wall}$, i.e., the blood velocity near the vessel wall is equal to the velocity of the vessel wall. The no-penetration condition on the other hand is defined as: $\mathbf{u} \cdot \mathbf{n} = \mathbf{v}_{wall} \cdot \mathbf{n}$. Thus, the blood velocity normal to the vessel wall is set to zero, whereas the velocity parallel to the wall is unaffected. The no-penetration condition is in accordance with our inviscid-flow modelling choice, where the effect of boundary layers is neglected.

Most of these common *Computational Fluid Dynamics* (CFD) methods require a precise modelling of the anatomy [28]. In some cases this can be very challenging, or even impossible to model the actual anatomy of the patient with current tech-



niques, e.g., when the anatomy is small in comparison with the imaging resolution, such as the heart valves [173]. To handle irregular solid-fluid boundaries, e.g., segmentations that do not align with the underlying grid, we use the variational solid boundaries approach by Batty et al. [34]. Here, the segmentation is represented as the zero level of a signed-distance field, i.e., for every voxel the smallest (signed) distance to the segmentation is stored. The gradient of such a field represents the direction to the nearest segmentation location. Fluid-cell masses, giving the weights in Equation (2.10), are estimated using a volume-of-fluid approach at non-grid-aligned vessel walls, which is computed based on the level set and the portion of the cell face between the cells that lays within the segmentation. The flow between fluid cells is determined by a normalizing these weights. The pressure solve uses these weights to constrain the flow to stay within the segmentation, i.e., it ensures the pressure solve produces a velocity field that matches the boundary condition. This means that the velocity component in the direction of the vessel wall should be zero (i.e., Neumann-type boundary condition). Therefore, by construction, this condition ensures no flow will cross the vessel walls. The segmentation is assumed to be static per measurement in our work, hence we compute these weights only once. More implementation details are given by Batty et al. [34].

6.2. Coupling

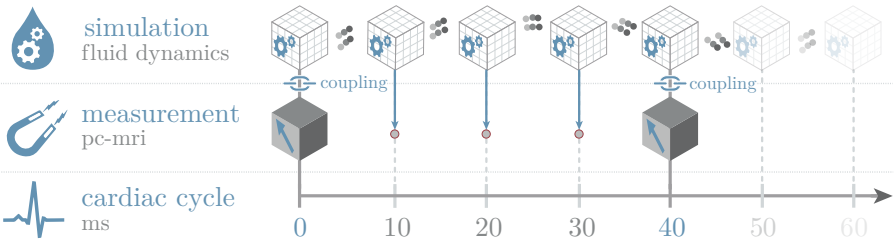


Figure 6.1: Overview of the measurement-simulation coupling. The 4D PC-MRI data is represented by cubes with an arrow. We propose a coupled approach between the full measured data and hybrid fluid simulation, represented by grids and particles. For each time point of the cardiac cycle at which PC-MRI data exists, the simulation is coupled with the measurements. In-between measurements, the simulation provides physics-based interpolated velocity fields, e.g., on the positions of the red dots.

An overview of our measurement-simulation coupling scheme is provided in Figure 6.1. Let $\mathbf{u}$ be the simulation velocity field, $\mathbf{u}_m$ denote the measured velocity field, and let $\mathbf{u}_d \equiv \mathbf{u} - \mathbf{u}_m$ be the velocity difference between simulation and measurements. Thus, ideally, in the absence of noise, $\mathbf{u}_d = 0$. The main idea of our coupling scheme is to construct the new simulation velocity field $\mathbf{u}^{new}$ at discrete time step $n+1$ as $\mathbf{u}^{new} \leftarrow \mathbf{u}^{n+1} - \gamma\, \mathbf{u}_d^{n+1}$, with γ a weight parameter and both fields $\mathbf{u}^{n+1}$ and $\mathbf{u}_d^{n+1}$ divergence-free, i.e., $\nabla \cdot \mathbf{u}^{n+1} = \nabla \cdot \mathbf{u}_d^{n+1} = 0$, see above. Although parameter γ can be used to bias selectively the resulting velocity field towards simulation ($\gamma = 0$) or measurement ($\gamma = 1$) across the time steps, we fix its value and

Algorithm 2: Measurement-simulation coupling.

1 Store current $\mathbf{u}^n$ on the grid, see step 3 of Alg. 1; the measured velocity $\mathbf{u}_m^n$ already comes on a grid;
2 Advect FLIP particles independently using $\mathbf{u}^n$ and $\mathbf{u}_m^n$, see step 4 of Algorithm 1;
3 Given $\mathbf{u}^*$ and $\mathbf{u}_m^*$, evaluate $\mathbf{u}_d^*$ on a grid;
4 Extend $\mathbf{u}_d^*$ and make it incompressible, see steps 5 and 6 of Alg. 1;
5 Extend $\mathbf{u}^*$ and make it incompressible;
6 Perform step 7 of Alg. 1 and store velocities $\mathbf{u}^{n+1}$ and $\mathbf{u}_d^{n+1}$ in the FLIP particles;
7 Set new simulation velocity using $\mathbf{u}^{new} \leftarrow \mathbf{u}^{n+1} - \mathbf{u}_d^{n+1}$;

use $\gamma = 1$. Note that the term $\gamma\,\mathbf{u}_d^{n+1}$ can also be interpreted as a spring force (or similarity term) with stiffness γ, keeping the simulation velocity field close to the measurement.

Since the measurement field represents blood-flow velocity samples in the main arteries, we assume that it also obeys the momentum-conservation law of the Navier-Stokes equations, i.e.,

$$\frac{\partial \mathbf{u}_m}{\partial t} = -\mathbf{u}_m \cdot \nabla \mathbf{u}_m. \tag{6.2}$$

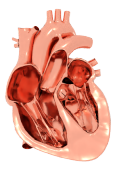
However, $\mathbf{u}_m$ may not be divergence-free, due to noise corruption and other scanning artefacts. Thus, unlike in the inviscid Navier-Stokes equations (2.7), we omit in Equation (6.2) the pressure term, enforcing fluid incompressibility. Subtracting Equation (6.2) from the corresponding Equation (2.7), one obtains

$$\frac{\partial \mathbf{u}_d}{\partial t} = -\mathbf{u} \cdot \nabla \mathbf{u} + \mathbf{u}_m \cdot \nabla \mathbf{u}_m - \frac{1}{\rho} \nabla p. \tag{6.3}$$

Like before, we rely on operator splitting to separate the advective and pressure terms from Equation (6.3). First we account for the $-\mathbf{u} \cdot \nabla \mathbf{u} + \mathbf{u}_m \cdot \nabla \mathbf{u}_m$ terms by advecting $\mathbf{u}$ and $\mathbf{u}_m$ forward in time using the FLIP scheme, giving a velocity difference $\mathbf{u}_d^*$. This velocity field is then made incompressible, similar to Equation (2.10), using the method by Batty et al. [34]. Once the pressure field is obtained, the Helmholtz-Hodge decomposition, or alternatively Equation (2.12), allows us to obtain the a field with near-zero divergence $\mathbf{u}_d^{n+1}$. To summarize, the main steps of our coupling scheme, applied whenever new measured velocities are available, are given in Algorithm 2. For the time steps when no measured data are available, the standard simulation algorithm, Algorithm 1, is used. The simulation grid size is set to the size of the input volumes containing velocity measurements.

6.2.1. Implementation

Our coupled simulation approach is implemented in the C++ programming language. For this, we extended the fluid solver by Batty et al. [34] to support our coupling scheme; see also Bridson [36] for additional implementation details regarding fluid simulations. For the involved linear algebra, we employ the LAPACK



library [174]. Furthermore, the visualization is created using the OpenGL graphics library, and we exploit the capabilities of modern consumer hardware using the GLSL shading language.

The current implementation preprocesses the coupled simulation results, providing new 4D velocity field with an increased temporal resolution. Computation time is in the order of a few minutes per time step of the measured data, depending on simulation grid size, the number of particles, and the complexity of the fluid behaviour. Many fluid simulations in the computer graphics research field, however, have been shown to perform in real-time on modern consumer graphics hardware, including the FLIP method we have adopted [175]. In contrast to CFD approaches, this provides great potential to perform a real-time coupling approach in new blood-flow visualization techniques.

6.3. Evaluation of coupling

In this section we present several experiments to assess the accuracy and robustness of the proposed coupling approach.

6.3.1. Experiment Setup

Validation of PC-MRI methods is difficult mainly due to the lack of ground truth. There is no analytical description for unsteady flows that goes beyond a simple pulsatile example in a straight tube [176]. Therefore, the generation of realistic synthetic data remains challenging. However, interesting general characteristics of the tested methods can still be analysed using such fields. The used synthetic data here consist of a parametric flow field describing a rotational vortex. The velocity $\mathbf{v} = (u, v, w)$ at position $\mathbf{x} = (x, y, z)$ is

$$\begin{cases} u(\mathbf{x}, t) = 2(10t + 1)(y - 0.5) \\ v(\mathbf{x}, t) = 2(10t + 1)(x - 0.5) \\ w(\mathbf{x}, t) = 0, \end{cases} \tag{6.4}$$

where t represents time, $t \in [0, 7]$. This simple field ensures a time-dependent velocity that increases linearly in time. The actual time between two consecutive outputs is $40ms$ and $x, y, z \in [0, 1]$. A boundary mesh, i.e., a cylinder, that matches the vortex, is also generated. The velocities at any position on the grid are known throughout time, and therefore are used as ground truth. Notice that lerp in time will give a perfect result for this simple data set.

For all experiments, two dissimilarity measures are used for the comparison of the synthetic ground truth of an experiment with the computed velocity field. Let $\mathbf{v}$ be the velocity field given by the ground truth and $\mathbf{u}$ the estimated/computed velocity field. We define two measures for comparison of the three-directional velocity data: the relative dissimilarity in magnitude δ_s, and the angular dissimilarity

measure, δ_a, i.e.,

$$\delta_s(\mathbf{x}) = \left| 1 - \frac{\|\mathbf{u}(\mathbf{x})\|}{\|\mathbf{v}(\mathbf{x})\|} \right| \tag{6.5}$$
$$\delta_a(\mathbf{x}) = \arccos\left(\frac{\mathbf{u}(\mathbf{x})}{\|\mathbf{u}(\mathbf{x})\|} \cdot \frac{\mathbf{v}(\mathbf{x})}{\|\mathbf{v}(\mathbf{x})\|} \right).$$

We separate the difference in speed and orientation, to get better understanding of the source of the differences. The ratio δ_s between the computed speed and the ground truth speed is such that 0 corresponds to $\|\mathbf{v}(\mathbf{x})\| = \|\mathbf{u}(\mathbf{x})\|$.

6.3.2. Coupling vs. Simulation

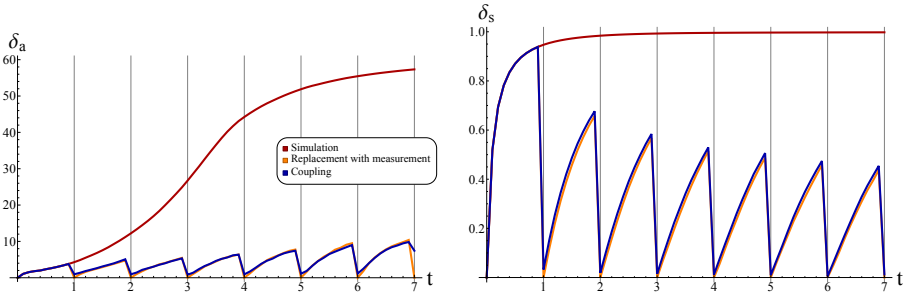


Figure 6.2: Comparison of the coupling with other simulation approaches using synthetic data. The field average of the per voxel speed and angle dissimilarity are presented.

The first experiment compares the accuracy of the coupling method with that of pure simulation. It is important to note that the simulation cannot mimic the synthetic data, because the speed in these data increases over time. Instead, the simulation speed dampens out over time. Therefore, the simulation error is expected to increase. For the evaluation, the synthetic data computed using Equation (6.4) will be used as measurement data. We compare the following approaches:

- Simulation: only the first measurement is used to initialize the simulation; the simulation will ensure that the flow field has near-zero divergence.
- Replacement-with-measurement: replaces at each time frame the velocity of every simulation particle with the velocity of the measurement at the particle position. In this case, the ground truth is better fitted, but the results are not guaranteed to be physically correct.
- Coupling: our newly introduced method should approximate the measurement, and at the same time, it should guarantee a flow field with near-zero divergence.

By definition, the replacement-with-measurement method is expected to give the lowest dissimilarity values, according to Equation (6.5). However, replacement-with-measurement produces a flow field that may not be physically correct, since the resulting flow may be compressible. Figure 6.2 shows the average over the field of the per voxel speed and angle dissimilarity of the different methods. A new measurement is applied when time t has an integer value, as represented by the vertical lines. The dissimilarity is measured every 0.10 time steps. Figure 6.2 shows that the coupling method, which ensures low divergent flow, has an error close to the replacement-with-measurement method, which is the lowest possible.

6.3.3. Noise Robustness

Magnetic Resonance Imaging (MRI) is subject to noise that influences the measurements, and results in uncertainty of the measured values. We expect that the coupling method, given noisy data, should produce more reproducible results than temporal lerp. In this section, the robustness to noise of standard interpolation and coupling methods is evaluated. The Rician noise of the PC-MRI methodology can be approximated by Gaussian noise [26]. For this experiment, different *Signal-to-Noise Ratio*s (sSNRs) are used. The SNR is defined as $SNR = P_s/P_n$, where P_s and P_n denote signal and noise power, respectively. P_s is given by the average velocity, whereas P_n can be set using the variance of the normal distribution [177]. Random values from a normal distribution are drawn using the Box-Muller approach.

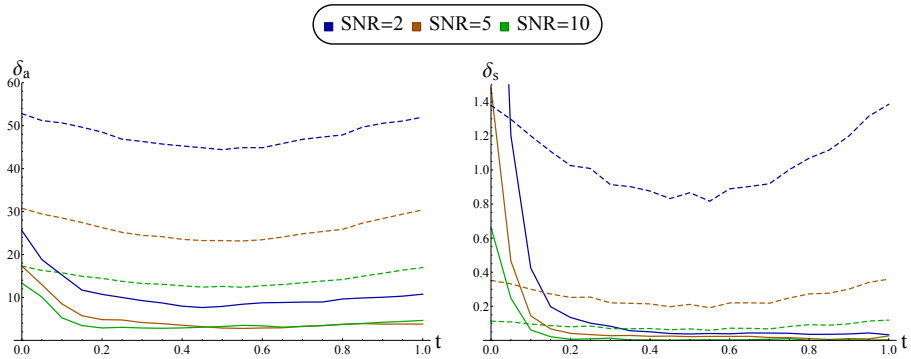


Figure 6.3: Comparison of the robustness to noise of the coupling method (solid lines) with the standard interpolation (dashed lines) using synthetic data. The average over the field of the per voxel speed and angle dissimilarity values for different SNR levels compared to the noiseless results are shown.

For this experiment, two simulations are run, using noisy and noiseless data respectively. Both simulations are initialized with the previously-defined synthetic data. From this data, the set with the highest speed is selected to initialize the coupling method. The simulations are run and compared until $t = 1$. In Figure 6.3, the per voxel average dissimilarity values using noiseless and noisy data are shown for different SNRs, namely 2, 5 and 10. Our measurements have an SNR of 10. As can be seen, in the coupling method the influence of noise is reduced. However,

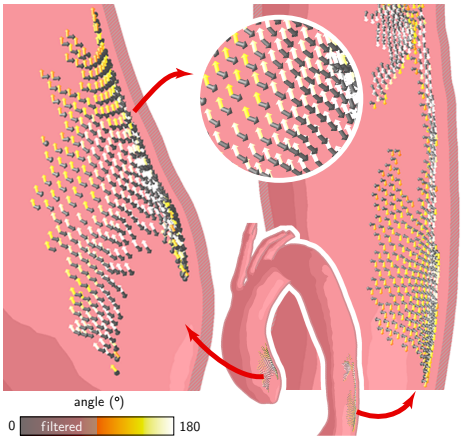


Figure 6.5: Measurement vectors are shown as grey arrows, and the coupled simulation vectors as arrows color-coded according to the angle between the vectors. Angles $> 90°$ are depicted. The vectors are located close to the boundary at peak systole for an aortic dissection case.

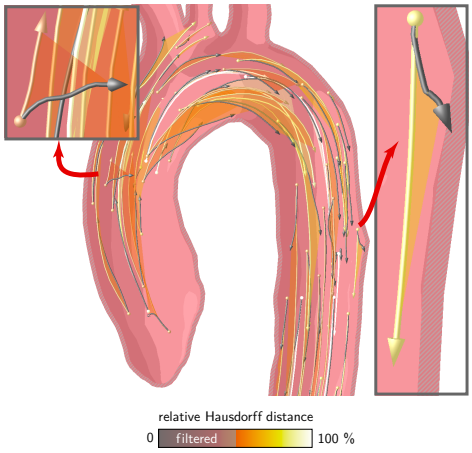


Figure 6.6: Comparative visualization of pathlines traced in the original measurements, depicted in grey, and the supersampled velocity field obtained by our coupled simulation. The simulation pathlines are color-coded according to the Hausdorff distance. Pathlines with the 50% largest distances are depicted for a healthy volunteer data set at peak systole.

when more initial noise is added (i.e., SNR=2), δ_a remains relatively high, compared to, e.g., the case with SNR=10. The results using lerp are also shown in Figure 6.3 right. In this case, noise has been added at the two time steps, and standard lerp has been applied for the positions in between. It can be observed that, in general, the robustness of the coupling method is clearly superior than that of interpolation. However, the coupling method is shown to be more sensitive to magnitude changes during the initial steps of the simulation when the influence of the measurements is the strongest.

6.4. Comparative Blood-Flow Visualization

In addition to the quantitative validation presented in the previous section, a qualitative assessment was carried out. Differences between the measurements and the coupled simulation are inspected using a dissimilarity measure, as well as tailor-made comparative visualizations.

We first inspect the dissimilarity measures at each time point of the blood-flow measurements. Figure 6.4 depicts the dissimilarity measures, defined by Equation 6.5, using a oblique slice at peak systole. Analysis of the dissimilarity measures reveals there are speed differences, in particular near the vessel wall. The overall speed, however, remains in the range of the measured velocity data. Angular differences occur in the top of the aortic arch, where the branching carotid arteries induce complex flow dynamics. The angular differences are, however, notably large near the vessel boundaries. Although for these data neither the measurement nor

the coupled simulation can act as a ground truth, the initial exploration indicates that the coupled simulation adapts the measurement, and likely corrects the velocity field, especially near the vessel wall. These regions are known to be susceptible to acquisition artefacts, mainly due to motion caused by the cardiac contraction [178].

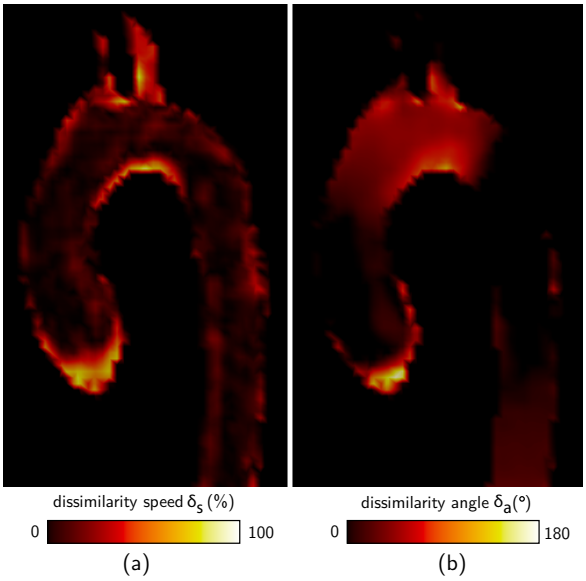


Figure 6.4: Results of the dissimilarity measures between the measured and coupled simulation velocities in an oblique slice. The slice is captured at peak systole for a healthy volunteer data set.

To further explore these differences near the vessel wall, we introduce a specific comparative visualization, as depicted in Figure 6.5. Based on seed points near the vessel wall, arrows are used to represent the velocity vectors of the measurements and the coupled simulation. The seed points are positioned at the inside of the vessel boundary by translating the mesh vertices inwards along the surface normal with a fixed small offset, i.e., 0.5 mm. Subsequently, velocity vectors at a relatively small angle, determined by the dot product, are excluded from the visualization using a user-determined filtering threshold. The arrows that represent the measurement velocities are conveyed in grey, while for the simulation arrows the angle between velocity vectors is color-coded using the blackbody radiation color map. To visually maintain the spatial relations, the arrows are embedded in an anatomical context, comprising a toon-shaded vessel wall surface rendering with its front-faces culled.

Figure 6.5 shows the comparative visualization of the velocity data at the aortic boundaries for a patient suffering from an aortic dissection. Some regions reveal strong differences in the measured and simulated velocity vectors. Closer inspection shows that the grey measurement arrows occasionally point retrograde to main blood-flow direction. This is unlikely at peak systole, also for pathological flow, indicating corrected motion artefacts. The fluid simulation does not enforce specific hemodynamics, and there is no explicit notion of antegrade flow: if the fluid physics and anatomical boundaries dictate retrograde flow, the simulation will yield accordingly, and so will our coupled approach. In this specific case, the coupled simulation vectors at these locations convey credible hemodynamics, deemed consistent with the physiology. This substantiates our supposition that the coupled simulation is

able to correct for acquisition artefacts near the vessel wall, based on the fluid physics.

Besides deviations near the boundary, we assess the differences between the full velocity fields using a comparative visualization based on integral lines. Therefore, we build on the work by Verma and Pang [179], which introduces a range of visualization approaches for flow data. In particular, we adopt the integral line comparison approach with a strip envelope.

Instead of streamlines, we compare pathlines that are randomly seeded throughout the aorta. At each seed position, two pathlines are generated for the duration of one time step in the measured data. The first pathline is traced in the measured velocity data, using Runge-Kutta 4 integration based on lerp. The second pathline is based on the coupled simulation, traced in velocity data that is supersampled in time. The increased temporal resolution affects the course of the simulation-based pathline, enabling comparison to conventional pathlines traced in measured data. The pathlines are represented by tubes with an arrowhead to indicate the direction, as depicted in Figure 6.6.

Similar to the comparative visualization at the boundaries, the pathlines based on the measurements are depicted in grey. The pathlines based on the coupled simulation convey the distance between the two pathlines, using a black-body radiation color mapping based on the Hausdorff distance metric. Using a user-defined threshold, pathline sets with a small Hausdorff distance can be omitted from the visualization. Figure 6.6 shows that the grey pathlines, obtained from the measurement data, exhibit aberrant behaviour near the vessel wall, due to the aforementioned acquisition artefacts. The zoom-in frames highlight two cases where the coupled simulation clearly adjusts the flow behaviour, yielding more plausible pathlines. Furthermore, the surfaces between the pathlines clarify that there are substantial differences between the pathlines within the bloodstream. The adjustments applied by the coupled simulation to the individual velocity vectors rapidly accumulate to considerably different pathlines. Since pathline visualizations are the prevailing visualization technique for blood-flow analysis, it is essential that the used velocity fields yields valid pathline representations. Our coupled simulation contributes by enforcing fluid physics.

6.4.1. Viscosity

A first attempt was made to extend the coupling method such that it also considers viscous effects. To achieve this, we included the additional viscosity term $\mu \nabla^2 \mathbf{u}/\rho$ and use Equation (2.5), with μ the dynamic (blood) viscosity. To be able to include the additional viscosity term $\mu \nabla^2 \mathbf{u}/\rho$ and solve Equation (2.5) the implicit viscosity solver by Batty and Bridson [35] can be used. Where μ is a constant representing dynamic (blood) viscosity. The method requires another solve of a symmetric positive definite linear system based on samples of the stress within the fluid. These samples are stored on the centres of the edges of the MAC-grid, and therefore, allows for a centred finite differencing of the adjacent velocity components. For more details, see the work of Batty and Bridson [35].

Similarly, the term $\mu \nabla \cdot (\nabla \mathbf{u}_m + (\nabla \mathbf{u}_m)^T)/\rho$ was included in Equation (6.2), since

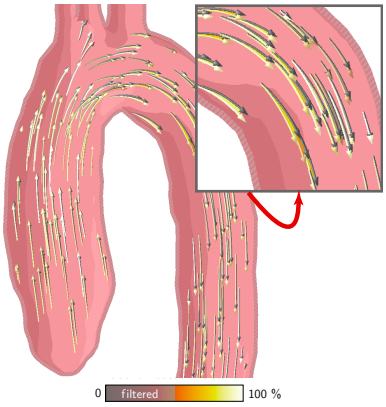


Figure 6.7: Comparative visualization between inviscid (grey pathlines) and viscid (pathlines color-coded by the Hausdorff distance) simulation velocities. In both cases, our measurement-simulation coupling scheme was used. Only the pathlines with the 50% largest distances are shown.

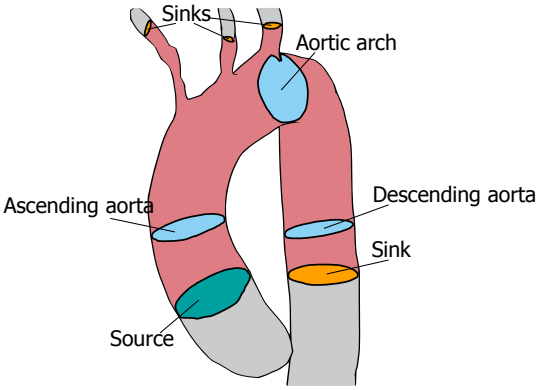


Figure 6.8: Using perpendicular disk-shaped sources (green) and sinks (orange), the user can select a valid region (pink area) in which to simulate. The lightblue disks indicate the used evaluation planes perpendicular to the mesh. These disks are placed in the ascending aorta, the aortic arch and the descending aorta.

we assume that the densities are identical and the measured velocity may not be incompressible, so that $\nabla \cdot (\nabla \mathbf{u}_m)^T \neq 0$. We then use the 'no-slip' boundary condition and the implicit viscosity solver by Batty and Bridson [35]. The result of a preliminary experiment, comparing inviscid and viscid simulation velocities, are shown in Figure 6.7. In both cases, the simulation velocities were coupled to velocity measurements. As can be seen, the differences in orientation and magnitude between the two velocity sets are small. In our experiments, we could also observe that the differences are larger close to the vessel walls, where viscous effects are expected to be larger. These initial experiments indicate that there are differences, and therefore the viscous model is preferable, since theoretically, it better represents physical reality.

6.5. Improvements

The previous method described above has a few drawbacks. For example, the temporal interpolation can be improved by taking both the previous and next measurement into account, opposed to only the previous measurement. To accomplish this, we simulate the flow from the next measurement backward in time. Furthermore, we counteract the damping effect that can be observed by introducing sources and sinks. To validate our method we use high-resolution measurements to and compare it with the previous approach, the current conventional interpolation scheme and measurements.

6.5.1. Backward simulation

The simulation is coupled with the measurements whenever measured data is available for that time step. Therefore, the simulation is steered when a measurement is available. In between the measurements, the simulation provides intermediate velocity fields based on an initial measurement until the next measurement and the simulation gets coupled again. Note that, due the the coupling, the near-zero divergence and no-slip boundary condition applies the measured time points and the intermediate velocity fields. However, when simulating only forward, the interpolation will be biased, towards one measurement in time, and not towards the two nearest measurements in time. This causes a discontinuity in time, when a measurement becomes available. To overcome this issue, we propose to also simulate backward in time.

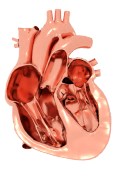
The Navier-Stokes equation for incompressible, inviscid flow is time-reversible [180], which is an essential requirement for our bidirectional coupling method. Indeed, since fluid viscosity is modelled as a diffusion process, i.e., a time-dependent process causing the momentum to change in space, time reversibility becomes an ill-posed problem. Many possible previous states exist, given an initial flow field, and as such, viscosity is omitted for backward simulation. Viscous effects are small for high-speed blood flow in big vessels for this reason and for simplicity viscosity was also neglected. However, close to the vessels walls viscosity is important. In order to still allow certain viscous behaviour of blood, we use the so-called 'no-slip' boundary condition here. This Dirichlet-type boundary condition states, for viscous fluids, that the velocity of the fluid at a solid boundary is equal to the boundary's velocity, i.e., $\mathbf{u} = \mathbf{u}_{solid}$. For static boundaries, $\mathbf{u}_{solid} = 0$, thus $\mathbf{u} = 0$ at solid boundaries. This approach results thus in an approximate viscous flow, without the need of modelling viscosity explicitly in the Navier-Stokes equation.

6.5.2. Bidirectional simulation

By simulating forward and backward in time two velocity fields are obtained for every point in time. Since both velocity fields are near-zero divergence, a weighted sum will result in another velocity field with near-zero divergence. For best results, this weight should be related to the distance in time to the last coupling of the simulation, forward and backward respectively. The longer the simulation runs without coupling, the more it is likely to deviate from the measured flow, hence the lower the weight should be.

A naive approach would be to use linear weighting, where a weight of one is assigned when the simulation is coupled and a weight of zero when another measurement is reached. However, using linear weights makes the weighting non-differentiable at the measurements, since the sign of the derivative is inverted. Hence, the derivative of the resulting velocity field over time, namely the acceleration, would not be continuous.

We use a smooth one-dimensional, fourth order Bézier curve with uniformly distributed control points, see Figure 6.9, which results in a C^1 continuous differ-



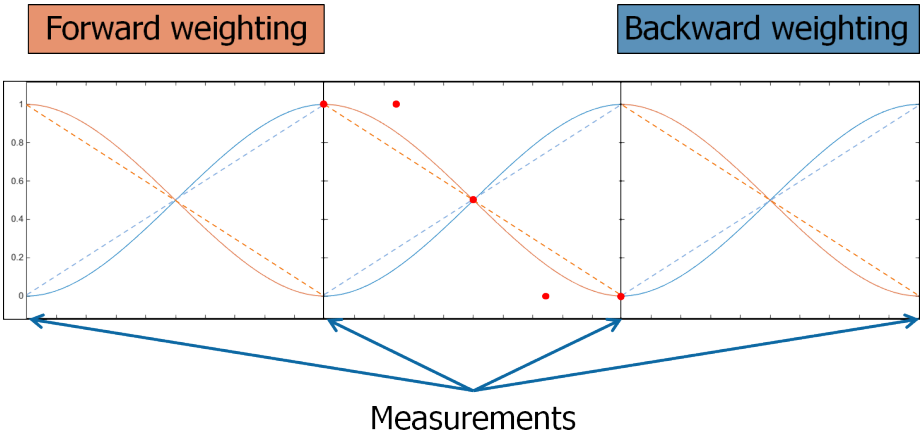


Figure 6.9: Using a smooth Bézier weighting (Equation (6.6)) for combining velocity fields with near-zero divergence, yields a new velocity field with near-zero divergence that is second-order continuous over time. The red dots indicate the 5 control points $(1, 1, \frac{1}{2}, 0, 0)$ for one of the fourth order Bézier curves ensuring C^1 continuity. The dashed lines indicate linear weighting, which is non-differentiable at the measurements.

entiable interpolation. After removal of the terms that evaluate to zero, gives us

$$B(t) = 3 \cdot t^2 \cdot (1-t)^2 + 4 \cdot t^3 \cdot (1-t) + t^4, \tag{6.6}$$

with $t \in [0, 1]$ is the normalized time between two consecutive measurements, where $t = 0$ is the current measurement and $t = 1$ represents the next measurement. The resulting curves, $1 - B(t)$ for forward and $B(t)$ for backward simulation, are shown in Figure 6.9.

These curves produce the weighting that result in the desired second-order continuous transition over time of the velocity field, opposed to using a linear weighting, while maintaining a total weight of one. Hereby, the acceleration is enforced to be smooth throughout the time domain. Another advantage is that the nearest measurement has a higher weight compared to linear weighting. It is to be noted that other strategies, such as B-spline or Hermite splines, that result in a smooth weighting could also be studied.

6.5.3. Sources and sinks

Simulating fluid in a closed container makes that the PDE for pressure has no solution, due to the so-called compatibility condition. Simulating in such a container is thus more difficult and requires assumptions about the fluid, which may induce compressibility. To avoid using such limiting assumptions, the previous approach removed particles within a certain distance to the grid boundary. This ensures that a free surface always exists close to the grid boundary, i.e., a surface separating fluid and air, as opposed to fluid and solid. Therefore, the polygonal mesh should be constructed, such that no fluid is desired to be simulated close to the grid boundaries, which limits the method to some extent. Furthermore, the amount of fluid

simulated could be undesirably large, resulting in a higher computation time. We provide more flexible user-definable sources, where fluid enters the system, and sinks, where fluid exits the system, such that the free surface is always available and the user can define where fluid should be simulated. The placement of the sinks and sources depends on the region of interest to ensure it is covered by the simulation.

Sources are grid cells that emit fluid continuously. The velocity of the particles in these cells is obtained from the measurements and using lerp in time. Notice that, also higher resolution 2D measurements could be used for this purpose. *Sinks* are cells for which all contained fluid particles are removed at every time step. Both, sources and sinks, shown in Figure 6.8, ensure that the fluid has a free surface, and thus, we do not simulate a fluid in a closed container. Note that the CFL condition, which limits a simulation time step to be such that a particle moves at most one cell per step, ensures that no particles overshoots a sink nor can a source be empty.

Fluid is initialized only in cells that are between a source and a sink cell, i.e., a cell contains fluid if and only if both, a source and a sink cell, are reachable without intersection of source, sink or solid cells –see the valid region in Figure 6.8. The only assumption is that the source and sink cells are always on a plane intersecting the solid, such that no fluid can leak through.

Finding cells, that should contain fluid, is achieved by using a sweeping algorithm. For every source and sink cell, a sweeping is started. The sweeping stops in a direction, if the cell reached is solid, source, sink or reachable by a cell with the same type, being either a source or a sink. In case a cell is reachable by another type, the cell is marked as valid and the sweep continues. The algorithm visits each cell at most three times, and can be performed as a preprocessing step after the source and sink cells are defined. Pseudo code for this sweeping algorithm is given in the additional materials.

Similar to the previous method, we add particles in empty cells that should contain fluid. If a measurement is available, the measured velocity is applied. Otherwise, we use the Fast Sweeping Method by Zhao [171] to extend the fluid velocity field to these empty cells.

6.6. Comparison

Due to the lack of a ground truth, validation and evaluation of the results is a challenge. Here, we will compare our results to a high temporal resolution PC-MRI acquisition. This data has a temporal resolution of 23ms, giving 46 phases for a heart cycle, while typically only 20 to 25 phases per cycle are acquired. We will also compare our results to previous methods: lerp and the previous method to analyse the differences.

For this purpose, we have to be able to compare two vector fields. Therefore, we developed multiple visualizations that help to analyse data sets both locally, as well as globally, in a qualitative and quantitative manner. We have multiple metrics on vector data sets, see Table 6.1 for an overview. Vector field $\mathbf{u}$ represents the given velocity field. Furthermore, $\mathbf{n}_s$ denotes the inward normal of the solid mesh,

required for the *Wall Shear Stress* (WSS) computation. Note that for an accurate calculation of WSS viscosity is relevant. However, an estimation of WSS is often calculated using PC-MRI measurements. An estimation of the viscosity is used as a constant value μ.

All these metrics represent different aspects of the flow that are of interest, either for blood-flow analysis (e.g. the WSS), or to show relevant general aspects of the flow. The *magnitude* of the flow on a plane is a representation of the amount of fluid that passes through that plane. It will, therefore, be used to compare the difference in flow, as well as in local speed. The *curl* of a vector field is the axis of rotation (vorticity). The magnitude of such a vector gives the local amount of vorticity. If there is curl at a point, the flow is not symmetric for that point. Globally, the more curl a flow shows, the more turbulent it is. *Acceleration* describes the change in velocity over time. Visualizing the acceleration helps to get more insight in the changes of the velocity field over time, which is important for understanding how a volume of fluid moves through the aorta, i.e., its advection. The *wall shear stress* (WSS) of a flow measures the shear stress the fluid exerts on the vessel wall. It is of clinical importance, since a high magnitude of WSS has been associated with cardiovascular diseases. Another interesting metric is the *divergence* of a vector field, which measures locally whether the vector field has sources or sinks apart from the input and output of the system, i.e., flow is respectively created or removed from the field. Note that for an incompressible fluid, like blood, the divergence should be zero, i.e., divergence-free. Thus, the flow does not contain sources nor sinks, excluding the ones defined by the user. In the simulated data near-zero divergence is imposed, so divergence will be close to zero everywhere. Yet, for the measured data it is known that this is not the case and it will not be zero everywhere.

Table 6.1: Different metrics used for analysis of the vector field $\mathbf{u}$.

Metric	Computation
Velocity magnitude	$\|\|\mathbf{u}\|\|$
Curl magnitude	$\|\|\nabla \times \mathbf{u}\|\|$
Acceleration	$\left\|\left\|\frac{d}{dt}\mathbf{u}\right\|\right\|$
WSS magnitude	$\mu\|\|\nabla\mathbf{u} \cdot \mathbf{n}_s\|\|$

Different qualitative visualizations are implemented to visualize the above metrics. One can select a disk intersecting the vessel wall to locally inspect the metrics. Furthermore, an iso-surface can be created to globally find regions that could be of interest, such as regions with high WSS. For direction and magnitude comparison of both velocity fields, one can visualize the pathlines and their distance when seeded from the same positions.

6.7. Evaluation

In this section, we validate our method by comparing our two methods to measured data and lerp. Due to the lack of a ground truth for PC-MRI data, it is difficult to validate any interpolation method. To circumvent this, we use both synthetic and measured data. The experiments considering robustness to noise are already

given. Our method does not add any variation to the analysis, therefore, robustness is omitted here.

6.7.1. Synthetic flow comparison

Due to the lack of analytical formulations for complex 3D flows, it is difficult to generate realistic synthetic flow data for the validation of measured flow. Yet, to test the interpolation, one can use synthetic data to analyst certain characteristics.

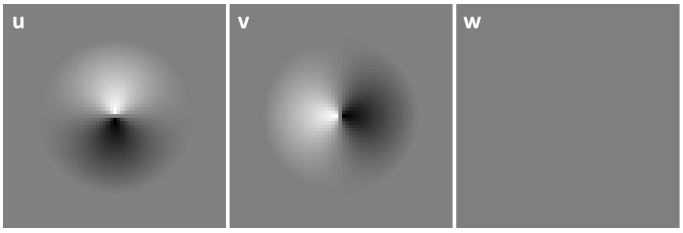


Figure 6.10: Three slices of the synthetic irrotational vortex flow in a sphere, showing the three velocity components (u, v and w) given by Equation (6.7).

Here, we use an irrotational vortex in a spherical mesh with increasing velocity over time $t \in [0,3]$ defined by velocity $\mathbf{u} = (u,v,w)$ for a position $\mathbf{x} = (x,y,z)$, where x, y and z are in the range $[-1,1]$:

$$\begin{cases} scale_r = \max(0.7 - \sqrt{x^2+y^2+z^2}, 0) \\ scale_t = t \cdot 20 + 40 \\ u^t(\mathbf{x}) = scale_r \cdot scale_t \cdot \dfrac{y}{\sqrt{x^2+y^2}} \\ v^t(\mathbf{x}) = scale_r \cdot scale_t \cdot \dfrac{-x}{\sqrt{x^2+y^2}} \\ w^t(\mathbf{x}) = 0, \end{cases} \tag{6.7}$$

where $scale_r$ scales the magnitude of the flow depending on the position within the spherical mesh and ensures that the velocity is $\mathbf{0}$ at the boundaries of the mesh, thus enforcing the no-slip condition; $scale_t$ is used to increase the flow magnitude over time. At the grid centre, the formula above is not valid, hence, we set the velocity to $\mathbf{0}$. Figure 6.10 shows three slices of the synthetic flow described above. For this synthetic data, the exact flow is known for every given point in time, hence a ground truth exists. Given the lack of a complex analytical flow description, in this special case, lerp would estimate the flow perfectly. However, the purpose of this analysis is to compare the general behaviour of our forward-only and bidirectional approaches. Figure 6.11 shows the average magnitude over time for the two methods. The forward-only method shows clear discontinuities in average magnitude values, with maxima just before being coupled to new synthetic-data samples (measurements). On the contrary, the bidirectional method circumvents this issue and shows much smaller deviations from the computed average magnitude. This example illustrates the theoretical advantages of using the bidirectional method, but does not show the characteristics of the method in real data.

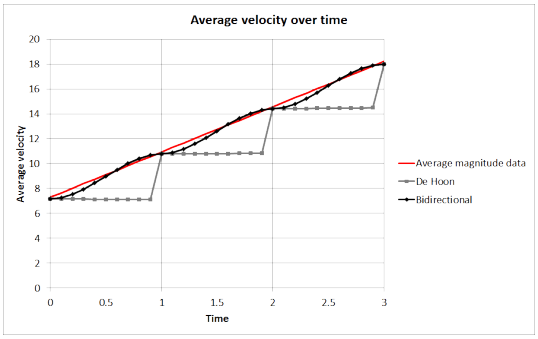


Figure 6.11: Using a synthetic irrotational vortex flow in a sphere with increasing magnitude over time, we compare the average magnitude of the flow for both our bidirectional method (black) and the forward-only method (grey) to the average magnitude cm/s computed from the data (red).

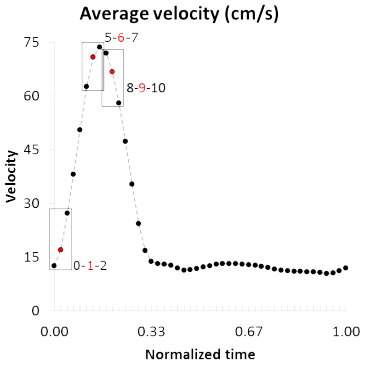


Figure 6.12: Every dot represents the average velocity inside the mesh per measurement. The time is normalized over a total of 46 measurements. The boxes represent the selection of measurements used per evaluation and the index of the selected measurements. The red dot represents the measurement that was left out, see text.

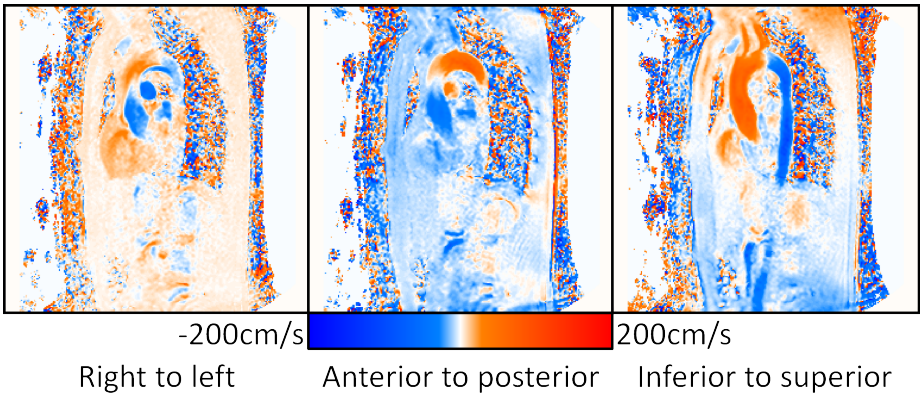


Figure 6.13: Three PC-MRI slices at peak systole showing the three measured velocity components. Blue and red respectively represent negative and positive values, while white indicates values near zero.

6.7.2. Measured flow comparison

We also evaluated our methods using reconstructed high temporal resolution PC-MRI data with no additional processing applied. Figure 6.13 shows three slices of this data set. The data has a higher temporal and spatial resolution compared to commonly used PC-MRI measurements, as one velocity component was obtained per acquisition, instead of all three components in a single acquisition. Consequently, the acquisition time is significantly higher, making it less suitable for clinical practice. 46 phases were measured covering a single heartbeat, while customary

PC-MRI measurements yield 20 to 25 phases per heart cycle. This high temporal resolution of 23ms is obtained on a Philips Ingenia MRI using the Turbo Field Echo Phase Imaging (TFEPI) protocol. The average measured velocity throughout the volume per phase is given in Figure 6.12. The three separate acquisitions of the velocity components resulted in three scalar volumes representing a single vector volume of $256 \times 256 \times 22$ voxels sized $1.5625 \times 1.5625 \times 2.5$mm. Measurements were performed with respiratory gating, a *velocity encoding speed* (v_{enc}) of 2m/s, a *Repetition Time* (T_R) of 5.807ms, an *Echo Time* (T_E) of 3.029ms and a *Flip Angle* (FA) of 10°. A mesh was derived from the *Temporal Maximum Intensity Projection* (TMIP) , using the Vascular Modelling ToolKit software (VMTK) [181].

For a qualitative and quantitative analysis of our method, we compare the result with the high-resolution measured data of a healthy volunteer. This *measured* flow is directly compared with lerp, our forward-only approach and our *bidirectional* approach. Removing a measurement allows us to compare interpolated data with the left-out measurement, while still maintaining a resolution close to common PC-MRI scans. We analyze both locally and globally the differences based on the metrics given in Table 6.1. Furthermore, we compare the volumetric flow rate through three slices shown in Figure 6.8, situated in the ascending aorta, the aortic arc and the descending aorta. We found that the divergence of simulated flow field is very close to 0, and therefore, a comparison of divergence will not bring much relevant information.

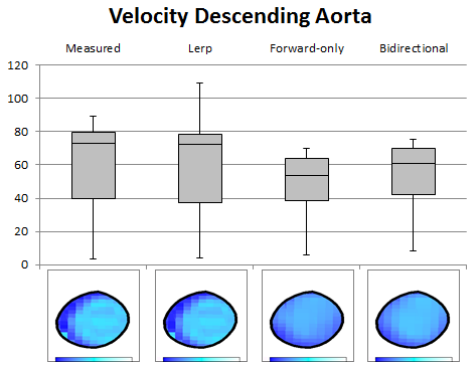


Figure 6.14: Box plots of the velocity in cm/s for the descending aorta and the corresponding cross section for each method are shown. The results for measurement 6 are shown. Both the measured and linearly interpolated cross sections show a high velocity in one of the lower left voxels, likely a consistent artefact. The coupled methods do not show a high velocity for this voxel. In each image the color coding encodes velocity magnitude in the range 0 to 200cm/s.

We focus on the systole, which is the period of high magnitude flow in the aorta due to the contraction of the heart, and hence has a high SNR. In Figure 6.12 the systole is the first $1/3$ of the cardiac cycle with a period of roughly 300ms.

Three subsets of the measured data were used, namely 0-1-2, 5-6-7 and 8-9-10 where respectively, measurement 1, 6 and 9 were left out for comparisons with the interpolation methods, see Figure 6.12. Other phases could be used, but these were selected since they represent the behaviour of the methods when dealing with increasing magnitude of the flow for different magnitudes. For decreasing magnitude of flow velocity, the methods behave overall similarly. However, when the damping of the forward simulation produces a result comparable to the measured flow, our method may yield a more deviating result, as shown in Figure 6.15. In this case, our method slightly underestimates the velocity magnitude by taking

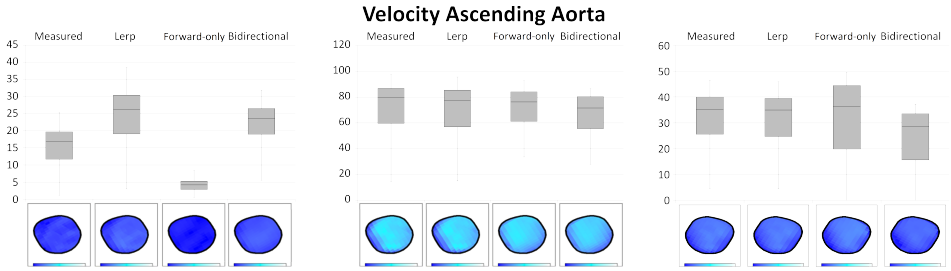


Figure 6.15: Box plots of the velocity in cm/s for the ascending aorta and the corresponding cross section for each method are shown. The results for left-out measurements 1, 6 and 9 (left to right) are shown. The velocity magnitude is similar for all methods, except the forward-only approach when the velocity is increasing, as shown by left-out measurement 1. Furthermore, the coupled methods show less variation and a more pronounced laminar flow pattern compared to the measured and linearly interpolation flow. In each image the color coding encodes velocity magnitude in the range 0 to 200cm/s.

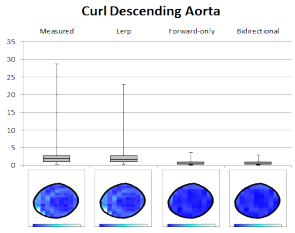


Figure 6.16: Box plots of the curl in cm/s for the descending aorta and the corresponding cross section for each method are shown. The results for measurement 6 are shown. For this measurement the flow is expected to be laminar, and thus, a low value or zero local curl is expected. In each image the color coding encodes curl magnitude in the range 0 to 15cm/s.

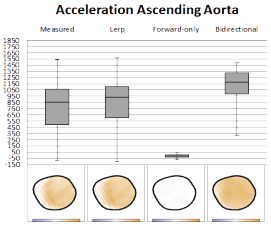


Figure 6.17: Box plots of the acceleration in cm/s^2 for the ascending aorta and the corresponding cross section for each method are shown. The results for measurement 1 are shown. The bidirectional method overestimates the acceleration, although it is closer to the measured acceleration than the forward-only approach. In each image the color coding encodes acceleration in the range -2000 to 2000cm/s^2.

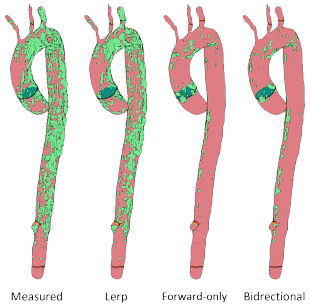


Figure 6.18: Iso-surfaces enclosing regions of high wall shear stress for left out measurement 6. In the simulated flow less regions are present, indicating a lower over WSS, amongst other causes, due to the implementation of the no-slip boundary condition.

measurement 10 into account.

For subset 0-1-2 the SNR is relatively low, since the flow magnitude is low, especially in the aortic arch and descending aorta. This results in more variation in the metrics, such as, the velocity and acceleration in the descending aorta. The coupled flow reduces this unexpected variation, however, since the flow velocity is increasing over time, the forward-only simulation does not interpolate the acceleration correctly due to the bias towards measurement 0. Figure 6.15 shows this bias is clearly present in the ascending aorta.

For subset 5-6-7, a clear laminar flow pattern is expected. The velocity of the flow is parallel to the walls and the speed varies from zero at the walls to a maxi-

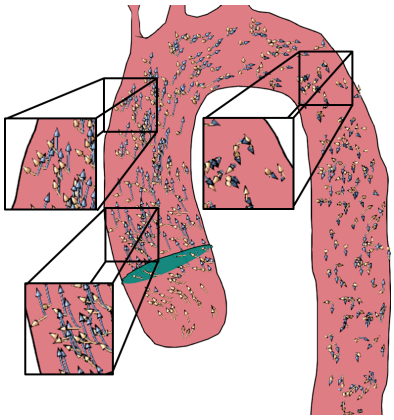


Figure 6.19: Flow comparison for left out measurement 1 in yellow and the corresponding bidirectionally simulated flow in blue. Measurement 1 has a low signal to noise ratio, this noise is less apparent in the coupled flow.

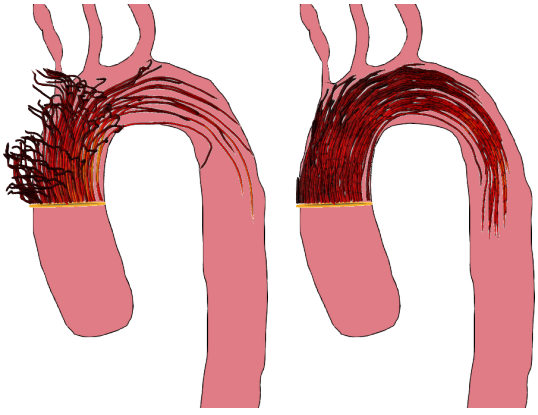


Figure 6.20: omparison of using lerp and our technique (right) for pathline visualization. The pathlines are seeded at the ring in the ascending aorta and were advected through the flow over time. The pathlines were computed using Runge-Kutta 4. Our technique provides a more laminar flow pattern and the effect of noise and artefacts is visibly reduced.

mum at the centerline of the vessel. This flow pattern is evident in the high-velocity measured flow, however, the coupled flow enhances it, as shown in Figure 6.15. Furthermore, consistent artefacts are corrected using the coupling, while they remain present in the linearly-interpolated result, as shown by Figure 6.14.

Overall, the coupling reduces the local curl significantly. This local curl represents local vorticity per voxel, and most likely is due to noise or artefacts. Especially in laminar flow, which has flow parallel to the vessel wall, low or zero curl is expected. An example of the reduction in curl by the coupling methods is shown in Figure 6.16. It shows the curl in the descending aorta for subset 5-6-7.

Our method still suffers from damping effects, although less, compared to forward-only simulation, as shown in Figure 6.14. Also, in some cases our method results in an overestimation of the acceleration, visible in Figure 6.17. However, it is clearly closer to the measured flow than the forward-only approach.

Figure 6.18 shows the WSS for measurement 6. aturally, by using the no-slip boundary condition, the simulation reduces the WSS. Furthermore, the estimated WSS of the measurements can partially be elicited by motion artefacts and segmentation errors, which are reduced by the coupled flow. Thus, the estimates for the WSS, are lower, however, the higher WSS is present in the expected regions, especially in the inner and outer arch.

Figure 6.19 shows the global flow for both linearly interpolated flow and bidirectionally simulated flow in subset 0-1-2, in which the SNR is low. Clearly, the measured flow is subject to noise and artefacts, which are less apparent in the coupled flow.

The effect of the interpolation on visualizations through time can also be seen in Figure 6.20. It shows the resulting pathlines for both linearly interpolated and cou-

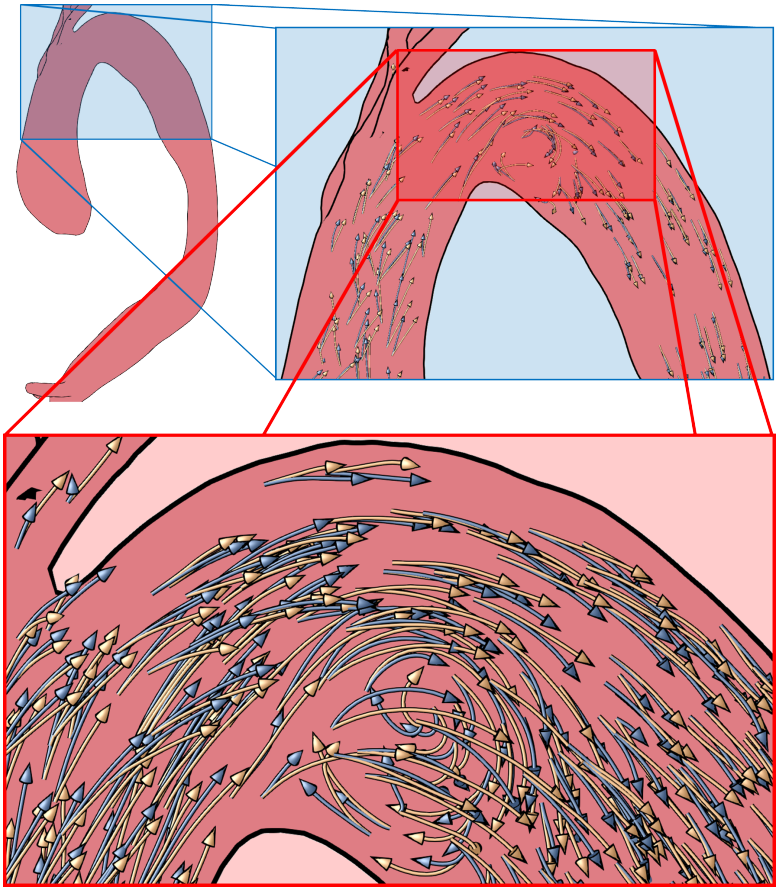


Figure 6.21: Important flow features, such as vortices, are maintained by our technique. Here, abbarent flow in a patient with an aortic dissection is shown. Yellow pathlines indicate the measured flow, while blue represents the coupled flow.

pled flow fields using Runge-Kutta 4 after a few interpolation steps. The influence of noise and artefacts is visibly reduced, furthermore, the flow has a more laminar pattern.

In Figure 6.21 the flow of a patient with an aortic dissection is shown. An aortic dissection occurs when a tear the inner layer of the aorta wall causes blood to flow between these layers of the wall of the aorta, resulting in a separation of the layers. In this case, both the flow and the anatomy deviate from the healthy case. A vortex forms in the aortic arch, Figure 6.21 shows its presence in both the measured and the coupled flow, demonstrating that our method preserves important flow features. Each of the 25 phases of aortic dissection data consists of a vector volume of $144 \times 144 \times 60$ voxels sized $2.0 \times 2.0 \times 2.5$mm. Acquisition was performed with a v_{enc} of 2m/s, T_R of 4.7ms, T_E 2.7ms, and FA of 5°.

6.8. Discussion

For the coupled simulation to converge, an appropriate initial condition is essential. Therefore, velocity data at time steps surrounding peak systole provide an adequate starting point. Noisy data with slow blood flow during early systole and throughout diastole will complicate convergence to a physically correct solution. Furthermore, the boundary conditions affect the results of the coupled simulation. The static manual segmentation of the vessel lumen best matches the bloodstream after the cardiac contraction during systole.

6.9. Conclusions and Future work

Blood-flow velocity measurements are prone to acquisition artefacts, while fluid simulations rely on uncertain model assumptions. We advocate a data assimilation approach that can benefit from both measurements and simulations, and subsequently, it can improve the visual and analytical exploration of hemodynamics. We have demonstrated a coupled fluid simulation approach that emanates from the full 4D PC-MRI blood-flow velocity data, while imposing physical properties of the hemodynamics. Based on the difference between the velocity fields of the measurement and simulation, the coupled simulation yields physically underpinned velocity data between the time steps of the MRI data, addressing the coarse temporal resolution of the measurements. Moreover, acquisition artefacts, in particular near the vessel wall, are corrected by the coupled simulation resulting in a more laminar flow pattern.

The proposed coupling method has been evaluated using synthetic data, investigating the accuracy and robustness to noise. The results indicate that the coupling method is more robust to noise than the standard interpolation method. By simulating flow in a bidirectional manner, forward and backwards through time, interpolation bias can evaded. The addition of sources and sinks, as well as the bidirectional simulation, reduces the simulation damping over time.

A visual assessment using the coupled simulation with 4D PC-MRI data showed that the largest differences are found near the vessel wall. The comparative boundary visualization revealed areas with strong angular differences between the measurements and the coupled simulations, where the adjusted velocity vectors of the simulation are in line with the known direction of the bloodstream. This inspires confidence in the physical underpinning of the coupled simulation. Further analysis with our comparative pathline visualization confirms this correcting behaviour near the vessel walls. Furthermore, this visualization shows that the pathline sets within the blood flow differ substantially, which is due to the line traces that accumulate the relatively small adjustments applied to the velocity field. This motivates the need for a combined approach to obtain the best possible velocity field as a basis for a pathline visualization. We further evaluated our method using synthetic flow data and high resolution measurements, to assess the interpolation quality, comparing for velocity magnitude, curl magnitude, acceleration and WSS magnitude. The measurements, albeit having a high resolution, were not of high enough quality

for a direct comparison since they still clearly contained noise and artefacts. The flow was shown to be more coherent and to have a more laminar pattern with less outliers than lerp. Furthermore, we have shown that using the bidirectional simulation approach is beneficial for the interpolation quality compared to forward-only simulation, which was shown to be robust to noise and artefacts. We also illustrated improvement on the visualization due to the interpolation. However, further validation, e.g., using a physical phantom, is necessary to provide solid conclusions. The use of more data sets would allow for statistic analyses of the results. Such validation is out of scope, but essential to prove the validity of the approach for clinical applications.

The presented approach has some shortcomings, for example, the local quality of the measurement is not taken into consideration. Moreover, the presented methods does not provide any control over the resulting velocity fields, hence, the field can have some undesired properties. For example, the velocity field can deviate too much from the measured data. In the next Chapter, these and other weaknesses are addressed.

To the best of our knowledge, we presented the first method that combines full 4D PC-MRI velocity data with fluid simulation. We underpinned the value of such a coupled approach, which is extended in the next Chapter.

7

Data assimilation

In this Chapter, we extent the previous Chapter by presenting a novel methodology using data assimilation techniques for PC-MRI noise and artifact removal. In contrast to the previous Chapter, we use an optimization approach that can be controlled by the user that can generate noise-free physically-plausible flow that is close to the measured data, whilst maintaining important flow features such as minimal divergence. Moreover, it also allows us to increase both the spatial and temporal resolution. To avoid sensitivity to the anatomical model, we consider and update the full 3D velocity field. To evaluate our method, we demonstrate our approach using phantom data with various amounts of induced noise. Furthermore, we show that we can improve the data while preserving important flow features, without the need of a highly-detailed model of the anatomy.

This chapter is based on the following publication:
Data Assimilation for full 4D Phase Contrast Magnetic Resonance Imaging measurements: Physics-based denoising and interpolation by N.H.L.C. de Hoon, A.C. Jalba, E.S. Farag, P. van Ooij, A.J. Nederveen, E. Eisemann and A. Vilanova in *Computer Graphics Forum (2020)* [182] © 2020 Wiley

Data assimilation can be used to estimate, interpolate, or extrapolate the true flow velocity field. The concept of data assimilation was also previously applied to *Phase-Contrast Magnetic Resonance Imaging* (PC-MRI) data. However, the methods are highly dependent on the model of the anatomy, such as the details of the anatomy, which limits their applicability and quality. This is because most of simulation models require a precise modelling of the anatomy [28]. In some cases this can be very challenging, or even impossible to model the actual anatomy of the patient with current techniques, e.g., when the anatomy is small in comparison with the imaging resolution, which is the case for example for the heart valves [173].

In this Chapter, we present a data-assimilation methodology that updates the full 3D PC-MRI flow data to be physically-plausible, with limited sensitivity to the accuracy of the given anatomical model and boundary conditions. To this end, we define an optimization process that minimizes the difference between the measured data and the output of a model based on the physical flow properties, which were given in Chapter 2.3. A quasi-Newton method [183, 184] is used to find the velocity field produced by the model that best corresponds with the measured data. To obtain the gradients necessary for the quasi-Newton method, we apply automatic differentiation [185] on the model code. Furthermore, this methodology can also be used to increase the resolution, both spatially and temporally, in accordance with the Navier-Stokes equations and the measured data. Our solution also allows including the concept of uncertainty, as discussed in Chapter 2.2.3. That is, local areas with reliable measurements have more influence on the resulting flow field, while the flow model has more influence in the areas where the measurements are less reliable. To the best of our knowledge we are the first to also consider the local quality of the PC-MRI data to steer the local weight of the measurement or the model in the data assimilation process. Moreover, previous methods do not consider temporal interpolation.

This Chapter is organized as follows, first we define the requirements for our data-assimilation methodology. After this, we present our methodology, which we then evaluate and show results. We conclude the Chapter and provide potential future work.

7.1. Requirements

The most important aspect of PC-MRI data is its *patient specificness*, however, the presence of noise and artefacts due to limitations of the scanning technique [2] make the data potentially not physically correct. For example, partial volume effects can occur due to the limitations of the measuring resolution. Therefore, the measured data near the vessel wall is often less trustworthy. The balance between a model and measurement data should be based on the local reliability of the data.

As indicated, for many blood-flow analyses and visualizations it is important for the flow to be *divergence-free* [97, 100]. The only sources and sinks of flow should be at the beginning and end of the vessel. This means that for a given velocity field $\vec{u}$, we want to ensure for every voxel that

$$\nabla \cdot \vec{u} = 0. \tag{7.1}$$

Another physical property is that no flow should leave the blood vessel, i.e., the vessel does not leak. Therefore, when flow is analysed or visualized all the flow inside a vessel should stay within the vessel. In other words, the velocity component in the direction of the vessel wall should be zero (a Neumann-type boundary condition). We consider the flow to be *physically-plausible*, if the flow is divergence free and no flow leaves the segmentation of the vessel walls. Furthermore, due to the relatively low spatial and temporal resolution of PC-MRI data, a *higher resolution* is beneficial for analysis.

In summary our methodology should generate flow data that fulfils the following requirements:

- *Patient-specific*: as close as possible to the original measured data where the measured data is reliable.
- *Physically-plausible*: divergence-free and no flow leaking through the vessel walls.
- *High-resolution*: enough data points to convey useful visualizations and analysis.

7.2. Measurement uncertainty

Due to noise and artefacts present in measured data, every voxel in the data set can have a different reliability ϵ. For an estimate of the voxel-wise *Signal-to-Noise Ratio* (SNR), and thus reliability, caused by PC-MRI measurement noise, we employ the method presented by Friman et al. [24, 25] which is explained in Chapter 2.2.3. This provides us with an initial value for ϵ. We then preprocess the data using the method proposed by Yang et al. [186] to remove severe artefacts. More specifically, we mark all voxels as invalid if any vector component deviates more than 25% from the weighted average of its surrounding voxels. By using a deviation of 25% the number of voxels that are marked invalid is small for our data sets, as indicated by Figure 7.1, if any voxels were marked as invalid at all. The resulting gaps are then filled with the average of the surrounding valid voxels, and the reliability ϵ of the corrected voxels is set to zero. Furthermore, we also mark all voxels that are likely to be affected by partial volume effects with reliability zero. That is, we update the SNR based on the distance to the segmentation boundaries. For example, we set the certainty to zero for all voxels that are crossed by the segmentation. This leads to reliability ϵ per voxel, where the value of ϵ is larger when the voxel is more trustworthy. Note that voxels with a zero reliability are only updated by the model considering the surrounding data. Meaning that the measured data stored in such a voxel has no impact on the optimization process, yet the resulting value is based on the surrounding, more reliable voxels. Therefore, the overall influence of these voxels on the resulting flow is negligible when the amount of voxels with zero reliability is low.

7.3. Automatic Differentiation

Three main techniques are commonly used to compute derivatives of a functional: *numerical differentiation*, *symbolic differentiation* and *automatic differentiation*.

Numerical differentiation, or finite differences computes the derivative of a function f' by evaluating the difference between $f(x)$ and variation(s) of the input. This, however, inevitably leads to numerical rounding errors. Furthermore, it requires the variation in input to be sufficiently small, and many function evaluations are needed when the number of input variables increases, e.g., when x is a vector.

Symbolic differentiation works by replacing every mathematical expression by its corresponding derivative. Symbolic differentiation is very difficult to perform correctly for complex expressions and requires the input function f to be presented as a mathematical expression, but allows for an analytical derivative. This requires careful consideration of all cases. Furthermore, the need for a single expression can become very challenging when f contains control flow statements such as **if** statements or **for**-loops.

Automatic differentiation, or algorithmic differentiation, is a collection of techniques that uses the chain rule to automatically compute the derivative of a set of mathematical operations. Automatic differentiation is based on the notion that every computer program executes a sequence of basic arithmetic operations. This may seem similar to symbolic differentiation, however, it mitigates the need for a single expression. This differs from symbolic differentiation, where an algorithm/function is treated as a single expression and the derivative is computed for the whole expression at once. Furthermore, automatic differentiation can handle control flow statements as long as they do not depend on the variables to be differentiated. Therefore, automatic differentiation is often used to compute the derivative of an implementation of an algorithm.

In this work we specifically use reverse-mode automatic differentiation, which is a form of automatic differentiation, that uses reverse accumulation to obtain the derivative of the original algorithm. That is, the original algorithm is evaluated starting at the outer function, while forward accumulation starts at the inner function. As an example, let us consider the following simple algorithm that computes $A(x) = \cos(x)^2$:

```
1: procedure A(x)
2:     y ← cos(x)
3:     return y²
```

For this simple example, the reverse-mode automatic differentiation would compute the derivative A' of A with respect to x as follows: Starting at first outer function, i.e., the power of two, we rewrite $A(x) = (y)^2$ or $A(x) = (F(x))^2$, where we abstract away the cosine using $F(x)$. Its derivative using the chain rule is then given by $A' = 2 \cdot F(x) \cdot F'(x)$, i.e., $A' = 2 \cdot y \cdot F'(x)$. The next step computes $F'(x)$; since $F(x)$ is a basic arithmetic operation, and we do not have to substitute it to compute

its derivative, so $F'(x) = -\sin(x)$. Then, the derivative is $A' = 2 \cdot y \cdot -\sin(x)$ and since y is $\cos(x)$ we get $A' = -2 \cdot \cos(x) \cdot \sin(x)$. This computation is called the adjoint code of A and provides an implementation to compute A'.

Note that the adjoint code uses the values of the intermediate dependent variables in reversed order, e.g., we needed the value of $y = cos(x)$ before we needed x. Typically these values are stored on a stack in memory when the original algorithm is executed.

Forward accumulation is also possible, but requires more storage and bookkeeping as it has to keep track of all intermediate variables and outcomes, and, as such, it is more memory intensive.

More details on automatic differentiation and on how to convert code into its corresponding adjoint code can be found in the additional material and in the work by Giering and Kaminski [187] and Hogan [185]. Note that it is possible to implement the adjoint code of an algorithm by hand [115]. However, this leads to two dependent codes that have to be maintained in parallel. To remove this dependency, we use the *Adept* [185] library for C++ to compute the derivatives of our pressure and advection solvers.

7.4. Data assimilation for PC-MRI data

In this section, we present a minimization framework that fulfils the requirements specified in Section 7.1. To model the behaviour of fluid, we use the Navier-Stokes equations as given in Section 2.3. Initially we omit the advection term for denoising. Note that, for denoising PC-MRI data this term is commonly omitted [37, 94, 98, 100]. Moreover, in this chapter we assume blood to be inviscid, i.e., we do not consider fluid viscosity. This simplifies the model and is a common assumption regarding blood simulation [37]. Moreover, for temporal interpolation our method requires simulating fluid with a negative timestep and only inviscid flow is shown to be time-reversible [180].

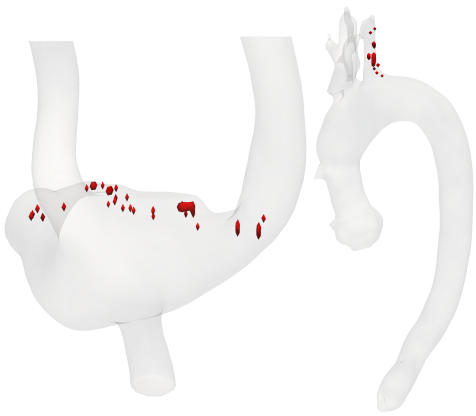


Figure 7.1: An overview the voxels that were marked as invalid during preprocessing.

The basis of our method is a function $P(\vec{v}) : S \subset \mathbb{R}^3 \to \mathbb{R}^3$, with S a subset in $\mathbb{R}^3$, that maps a velocity field $\vec{v}$ to another velocity field $\vec{y}$. For denoising, P can be a divergence-free filter performing a pressure projection as given in Chapter 2.4.1, which solves the pressure and divergence terms of the Navier-Stokes equations. Note that if the given velocity field is non-smooth, the pressure solve may fail to correct for its non-zero divergence, hence, a sufficiently smooth input velocity field

is necessary. To handle irregular solid-fluid boundaries, we use the approach by Batty et al. [34] as explained in Chapter 6.1.1.

7.4.1. Minimization

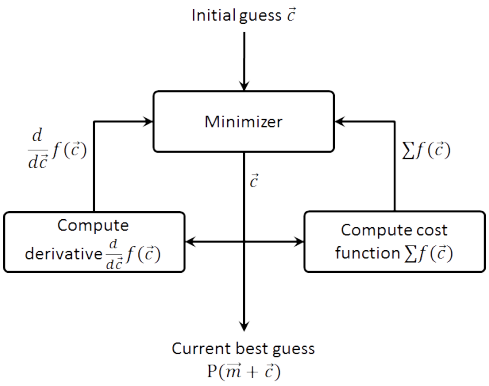


Figure 7.2: A schematic overview of our framework.

Using the pressure solve function P described previously, we can find physically-plausible velocity fields. However, these fields are not necessarily as close as possible to the measured data. Therefore, our goal is to find a suitable input for the function P such that the field produced by P is as close as possible to the measured field $\vec{m}$. To achieve this, we use a least-square optimization with constraints. To ensure similarity to the measurements, we use the following squared least error functional,

$$\underset{\vec{v}}{\operatorname{argmin}} \left(\quad \alpha \cdot \epsilon \cdot \left\| P(\vec{v}) - \vec{m} \right\|^2 \right), \tag{7.2}$$

where $\vec{v}$ is the field that we solve for, $\alpha \geq 0$ is a constant weight, and ϵ provides the local reliability for every voxel as described in Section 7.2. Please note that we are actually interested in $P(\vec{v})$, as discussed above. Note that this cost function corresponds to the *3-Dimensional Variational Assimilation* (3D-VAR) cost function with a perfect model assumption. Furthermore, it is similar to the cost function used by Bostan et al. [99] without their regularization. A schematic overview of the minimization process is shown in Figure 7.2.

Note that the function P could be replaced by the full Navier-Stokes equations that include the temporal evolution of the velocity field. However, since the temporal resolution of the PC-MRI data is low, we omit the temporal element of the velocity evolution that is part of the Navier-Stokes equations. Moreover, experiments using such a function that includes the temporal component and using a steady flow assumption, similar to the approach by Rispoli et al. [127], led to results that deviated more from the measured data.

To constrain the minimization, we rewrite Equation 7.2 as a control problem, comparable to McNamara et al. [115]. That is, we substitute $\vec{v}$ by $\vec{m} + \vec{c}$, so that the goal is to find the control vector field $\vec{c}$ that corrects the measured field $\vec{m}$, it is as small as possible, and yields $P(\vec{m} + \vec{c})$ that is close to $\vec{m}$. The corresponding minimization is then

$$\underset{\vec{c}}{\operatorname{argmin}} \left(\begin{array}{l} \alpha \cdot \epsilon \cdot \left\| P(\vec{m} + \vec{c}) - \vec{m} \right\|^2 + \\ \gamma \cdot \epsilon \cdot \left\| \vec{c} \right\|^2 \end{array} \right), \tag{7.3}$$

where the second term, weighted by $\gamma \geq 0$, is a Tikhonov-style regularization term

that penalizes too much control, so that changes to the measured data are kept to a minimum. Currently, our approach does not specifically avoid potential local minima. However, since the minimization is initialized close to the measured data, when the minimization finds a local minimum the result should be relatively close to the measured, target, data.

Applying Helmholtz-Hodge decomposition assumes that the input vector field is sufficiently smooth [101] which may not necessarily be true for the measured data due to the presence of artefacts and noise. Therefore, it is possible that not all divergence can be removed by the pressure solve. Furthermore, the minimization and the pressure solve P have conflicting goals: the minimization tries to keep the solution close to the measured data, while the pressure solve deems it physically plausible. Therefore, in order to decrease the cost function, the minimization can produce non-smooth fields for which the pressure solve cannot correctly compute a smooth pressure field. To address this issue, we include an additional term that punishes the minimizer for producing fields that the pressure solve cannot project. More specifically, this term penalizes the minimizer when it produces fields with a high divergence. Therefore, this term is given by the squared divergence, i.e.,

$$\underset{\vec{c}}{\operatorname{argmin}} \left(\beta \left(\nabla \cdot P(\vec{m} + \vec{c}) \right)^2 \right), \tag{7.4}$$

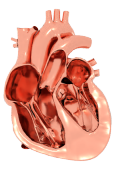
where $\beta \geq 0$ is a constant weight. Note that here also the divergence computation is weighted using the variational solid boundaries approach by Batty et al. [34] to ensure the flow does not leave the segmentation.

Therefore our final cost functional f is a combination of Equations 7.3 and 7.4, so that the optimization problem becomes

$$\underset{\vec{c}}{\operatorname{argmin}} f(\vec{c}) = \underset{\vec{c}}{\operatorname{argmin}} \begin{pmatrix} \alpha \cdot \epsilon \cdot \left\| P(\vec{m} + \vec{c}) - \vec{m} \right\|^2 + \\ \beta \left(\nabla \cdot P(\vec{m} + \vec{c}) \right)^2 + \\ \gamma \cdot \epsilon \cdot \left\| \vec{c} \right\|^2 \end{pmatrix}, \tag{7.5}$$

where α, β and γ are user-set parameters, and ϵ is defined by the data and depends on the measurement field $\vec{m}$.

Since the cost function is non-linear, we use a quasi-Newton type minimization, more specifically, we use the *Limited-memory Broyden-Fletcher-Goldfarb-Shanno* (L-BFGS) algorithm [183, 184]. L-BFGS iteratively finds a minimizer by evaluating the cost function and its derivative. That is, the control vector field $\vec{c}$ provided by the minimizer is used to compute the cost functional and its gradient. The minimizer then uses this information to improve the control vector field $\vec{c}$, so that $P(\vec{m} + \vec{c})$ becomes the current approximation of the optimal physically-plausible field that matches the measured field $\vec{m}$ as closely as possible. We set the algorithm to iterate until either the gradient value drops below a threshold, or the value of the cost functional did not decrease sufficiently in the last few iterations. The computation of the derivative of the cost function is done using automatic differentiation via the adjoint code of the implementation, see Section 7.3. A schematic overview of the minimization process is shown in Figure 7.2.



7.4.2. Spatial interpolation

To increase spatial resolution, we can use the system to interpolate physically-plausible values between the known values. That is, when upsampling, we consider the information between the measured voxels to be missing data. Therefore, we double the spatial resolution and linearly interpolate between the known voxels to get a new measured field $\vec{m}$. Since the interpolated voxels are most likely incorrect, we set their reliability ϵ to zero. That is, no matter what value they have, the minimizer will only take them into account for the divergence term given in Equation 7.4. Also the pressure solve does take them into account and will change them to make the field divergence-free. Since the minimizer will try to reduce the error with respect to the measurements, the resulting field will be as close as possible to the measured data. Both the minimizer and pressure solve will change the interpolated voxels to best fit the measured data and have a physically-plausible result. Therefore, the spatially-interpolated data fulfils all the requirements of Section 7.1.

7.4.3. Temporal interpolation

In order to perform temporal interpolation, we need a cost function that considers the temporal aspect of the data. Therefore, we need a model that describes how the flow evolves over time.

Simulation model

In order to simulate the behaviour of fluid over time the Navier-Stokes equations should be solved. Therefore, since we already have the pressure solve, we only require an advection scheme. That is, we need a model for velocity transport by the velocity field through time, i.e., prescribe how the velocity field moves through time.

To this end we use the semi-Lagrangian advection method introduced by Stam [27]. This means the velocity field $\mathbf{u}$ is updated for every grid position by tracing backwards the path of a *virtual* particle. That is, for a grid position p and time step Δt the new velocity is given by: $\mathbf{u}_{new}(p) = \mathbf{u}(q)$, where q is the backtracked position of the virtual particle, i.e., the location where the particle was at time $t - \Delta t$; hence $q = p - \Delta t\, \mathbf{u}(q)$. To avoid flow leaving the segmentation, we again constrain the flow using the same approach as before which is described in Chapter 6.1.1.

This method is stable, and furthermore, due to the iterative nature of the minimization, the simulation will not suffer much from numerical dissipation, since it will be limited to a single time step. Of course, more accurate hybrid schemes such as the fluid implicit particle method *Fluid Implicit Particle* (FLIP) [34] perform in general better than this simple scheme. However, a non-trivial conversion between particles and grid is needed for the derivative calculation. This makes FLIP not a suitable method and would also make it more computationally intensive.

Minimization

In order to have a physically-plausible temporal interpolation, we search for a physically-plausible field $\vec{y}$ at time $t + \frac{1}{2}\Delta t$, which is in between two measured

velocity fields $\vec{m}_t$ and $\vec{m}_{t+\Delta t}$ at time step t and $t+\Delta t$, respectively. Note that this requires simulation backwards in time, that is, we simulate from $\vec{y}$ at time $t+\frac{1}{2}\Delta t$ to $\vec{m}_t$ at time t. Since we have an inviscid fluid, the negation of the velocity field is equivalent to using a negative time step [180].

Given a simulation function $S(\vec{v}_t, \Delta t)$ which evolves $\vec{v}$ as in Subsection 7.4.3 over a time step Δt, pressure solve function $P(\vec{v}_t)$ and input velocity field $\vec{v}_t$ at simulation time t, we can setup the following cost functional

$$f(\vec{c}) = \begin{pmatrix} \alpha \cdot \epsilon_1 \cdot \left\| S(P(\vec{m}_i + \vec{c}), -\frac{1}{2}\Delta t) - \vec{m}_t \right\|^2 + \\ \alpha \cdot \epsilon_2 \cdot \left\| S(P(\vec{m}_i + \vec{c}), +\frac{1}{2}\Delta t) - \vec{m}_{t+\Delta t} \right\|^2 + \\ \beta \left(\nabla \cdot P(\vec{m}_i + \vec{c}) \right)^2 + \\ \gamma \cdot \epsilon_i \cdot \|\vec{c}\|^2 , \end{pmatrix}. \tag{7.6}$$

where $\vec{m}_i = (\vec{m}_t + \vec{m}_{t+\Delta t})/2$ is the linearly interpolated field of the two measurements and ϵ_1, ϵ_2 are the per-voxel reliability for the two measurements and ϵ is the interpolated reliability per voxel, i.e. $\epsilon_i = (\epsilon_1 + \epsilon_2)/2$. This is used as the initial guess for which we want to determine the control field $\vec{c}$. Furthermore, we use α as the user-given weight to both measurements, since the temporal distance between the two measurements is equal. After the minimization of $f(\vec{c})$, the resulting physically-plausible field is given by $P(\vec{m}_i + \vec{c})$. Note that his formulation is similar to Equation 7.5, albeit the first least-squared error term is replaced by two least-squared terms, after simulations with the nearest measurements, $\vec{m}_t$ and $\vec{m}_{t+\Delta t}$. Since both measurements are not necessarily noise free, they can be denoised beforehand using the method described in Section 7.4.1.

7.5. Results and evaluation

In this section, we present results to illustrate our method and evaluate it based on the requirements given in Section 7.1. The requirement of yielding physically-plausible fields is intrinsically satisfied by our method since the physcial properties of the flow are part of the model. However, we also evaluate several aspects, such as to which extent the field is divergence free. We also compare our approach to other state of the art methods. For the visualization of the flow we use streamlines, which is commonly used with PC-MRI flow data [2, 11, 17, 18]. The flow is unsteady, so the data consist of multiple time steps, and as such, pathlines would be suitable. However, the time step between the phases is relatively large in relation to the flow speed, and therefore, often short streamlines are used. Streamlines also illustrate the changes made to the flow field for a single phase, allowing us to compare individual phases. In all our experiments, the conjugate gradient method used by the pressure projection step converged in less than 20 iterations with a tolerance factor of 1×10^{-4}.

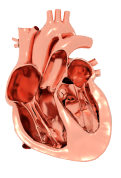


Figure 7.3: Streamlines of the measured flow before our method was applied. All visualization parameters (number of lines, seeding position and cutting plane) are equal to those in Figure 7.4. Note the presence of a vortex in the top-right corner and the helical flow present in the top-left corner and from the centre towards the top-right corner.

Figure 7.4: Streamlines of the measured flow after our method was applied. All visualization parameters (number of lines, seeding position, cutting plane and camera parameters) are equal to those in Figure 7.3. Note that the flow features are preserved while more streamlines are visible due to the lack of divergence and flow not leaving the segmentation.

7.5.1. Evaluation data

There is no trivial ground-truth for blood-flow data. Ideally, one would use a ground truth for validation, e.g., an analytical solution. However, while some analytical solutions to the Navier-Stokes equations exist for specific flows, they do not exist for non-trivial flows that possess the characteristics of blood flow [176]. As a result, high-quality experimental data is often used for the validation of both simulations and new measurement techniques. One way to obtain such experimental data is by the use of a physical phantom, as detailed in Chapter 2.2.4.

For the evaluation of our method, we use 4D PC-MRI measured data of a glass phantom that was created based on a 3D scan of an intracranial aneurysm in a patient. A pulsatile (time-varying) flow was generated based on the velocity profile measured in the internal carotid artery of the patient. More details regarding the acquisition of this type of data can be found in the work by van Ooij et al. [188]. One of the advantages of phantom data over patient data is that the flow can be measured at a higher resolution and with a higher SNR, since the imaging time can be increased. Therefore, phantom data has a relatively low amount of measurement noise. Furthermore, using a glass model the vessel wall is well defined and hence less (motion) artefacts can be expected. The measured phantom data has a resolution of $80 \times 47 \times 51$ with a voxel size of $0.2 \times 0.33 \times 0.2$ millimetres. Flow was color-encoded with flow speeds between 0 and 60 cm/s. The measured data is shown in Figure 7.3 without any processing which is common current practice.

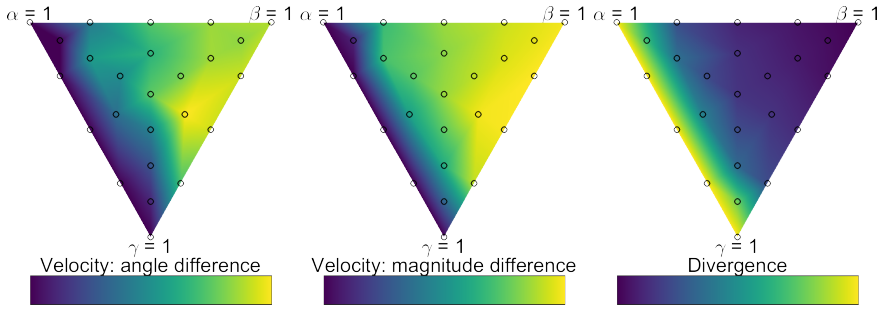


Figure 7.5: The sensitivity of our method for different ratios of the three parameters α, β and γ. The two left-most triangles show how well the velocity vectors match the measured input. The third right-most triangle shows the remaining divergence. colors represent the average value for the tested configuration. Note that darker colors mean the method performs better. The lowest and highest values have been clamped to show a better representation.

Figure 7.4 shows the results after applying our method. The results of the evaluations are gathered in Table 7.1. A visual comparison of the various input fields and resulting flow fields is shown in Table 7.2. For all visual comparisons exactly the same seeding positions, and the same slime of the velocity magnitude is shown. To illustrate the results and preservation of features with real data sets, several results 4D PC-MRI data sets will be shown.

7.5.2. Parameter sensitivity

Our method relies on three weighting terms, α, β and γ, the ratio between these weights determines their importance for the cost function, and thus, influences the outcome. The sensitivity of our method to different ratios of the weights is shown in Figure 7.5. The circles represent the sampled positions, where every sample represents a ratio where the total sum of the parameters is 1, e.g. $\alpha + \beta + \gamma = 1$. This means that in every corner of these equilateral triangles one of the parameters has value 1 and the others are 0. We evaluated how close the result was to the measured input by comparing the average of all velocity vectors, i.e., the average difference in velocity magnitude and the average difference in angle is shown. Furthermore, the average of the absolute divergence that is still present in the data is shown. Since the results vary a lot, the highest and lowest values were clamped. From the evaluation it follows, as expected, that the terms α and γ ensure the result is close to the measured data. Naturally, the term β leads to a more divergence-free field. If the term $\beta = 0$, the remaining divergence is relatively high, but the result is closer to the measured data. Based on this sensitivity analyses, throughout the paper, we use the following values for the weighting ratio of the terms : $\alpha = 0.4$, $\beta = 0.4$ and $\gamma = 0.2$. These values are close to the optimal shown in the Figure 7.5. While they are not the exact optimal values, this ratio seems to achieve the best trade-off between closeness to the measured data and remaining divergence. However, this setting is not critical as the influence of the parameters seems to be rather stable in our empirical analysis. Moreover, the optimal ratio

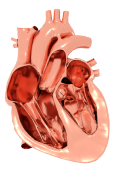


Table 7.1: The mean and standard deviation of the absolute differences in magnitudes and angles given by |base case – result|.

	Magnitude difference		Angle difference	
	Mean	std dev	Mean	std dev
Base case	1.66	4.77	14.58	20.12
SNR 10	2.43	2.68	14.77	21.63
SNR 4	2.43	3.03	17.66	22.19
Spatial interpolation	1.37	3.21	14.13	19.27
Temporal interpolation	1.64	1.89	11.61	19.44

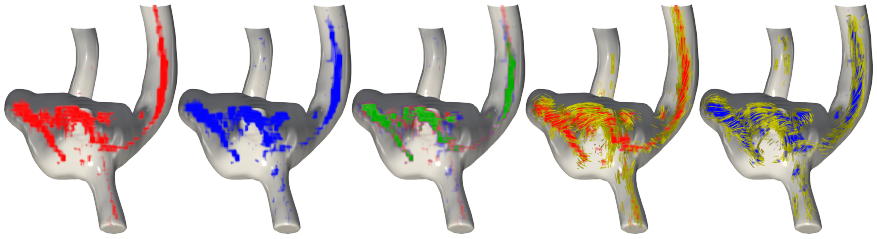


Figure 7.6: Based on the the λ_2-criterion vortex cores can be found. Using volume rendering the vortex cores are shown for the original measured data (red), the data after our approach was applied (blue) and the union of the two volumes (green). The most right two images show the corresponding streamlines seeded from the vortex cores of the measured data and the data after our approach, respectively.

deviates slightly for every measurement, and hence, would have to be determined per measurement.

Figure 7.3 shows the results of visualizing the measured data obtained from the scanner without any processing. Figure 7.4 shows the results after applying our method using the settings mentioned above. We can observe that, after our method was applied, the streamlines do not merge due to divergence and more streamlines can be traced for a longer amount of time without leaving the segmentation. The same features are visible in both figures, although they are much clearer after applying our method. Note that streamlines were seeded with the same settings for both figures.

7.5.3. Denoising

The evaluation of the denoising capabilities of our method consists of two parts. First, we show the robustness of our method to various amounts of noise, followed by a comparison with previous methods.

Robustness

To test the robustness of our method to various levels of noise, we need to control the amount of noise present in the data. The denoised data, shown in Figure 7.4, will be used as ground truth for our next evaluation step. For this, we add various amounts of noise to the data. This way we can compare the result of our denoising

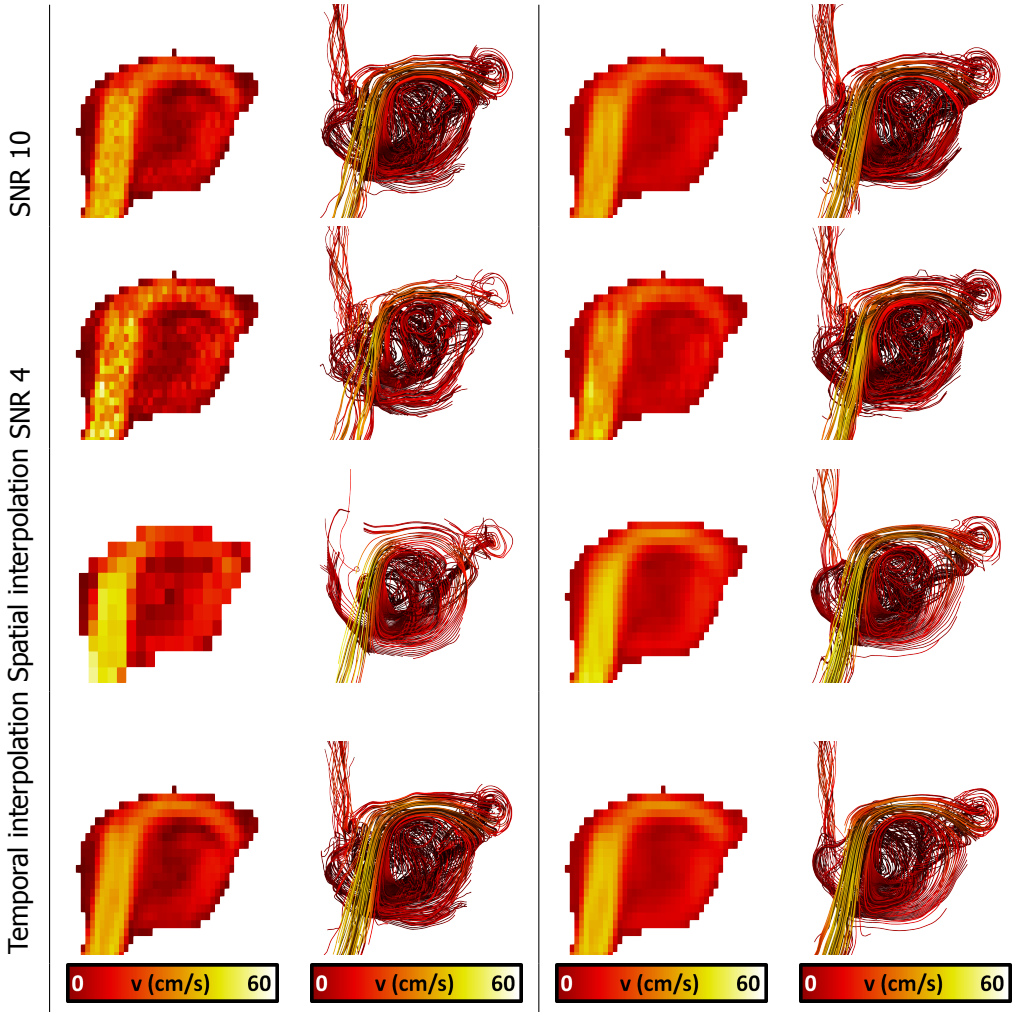


Table 7.2: A visual comparison of our method when used for denoising and interpolation. The velocity magnitude of a fixed slice through the volume is shown. For all streamline images, the same seeding positions and same number of streamlines are used. The input data is shown on the left, while the result of our method is shown on the right. The reference for the streamline images is shown by Figure 7.4 Note that, despite the different input scenarios, the images on the right are comparable. Moreover, the flow features as visualized by the streamlines in the reference image are also present after out approach was applied.

approach with the expected noise-free output. This allows a direct comparison of the results and helps us determine how much and where the flow field is modified. Note that we do not use the original measured data as shown in Figure 7.3, because there is noise present in the data which would make it very difficult to determine the robustness of the method.

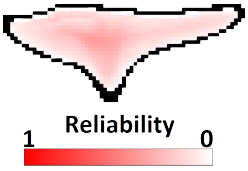


Figure 7.7: Slice through the volume showing the reliability per voxel of the measured data. Note that the reliability is lowest near the vessel wall.

To mimic various signal-to-noise ratios we add Gaussian noise using the following function

$$\vec{u} = \vec{u} + G(0, \frac{1}{SNR} \cdot \|\vec{u}\|), \tag{7.7}$$

where $G(\mu, \sigma)$ draws a sample from the Gaussian distribution with mean μ and standard deviation σ. This is done separately for every velocity component based on the signal strength $\|\vec{u}\|$ weighted by the desired SNR. Therefore, the average velocity information per voxel will be $\frac{1}{SNR} * 100\%$ added noise. Note that, the lower the SNR the more noise we add. Results are given for SNRs of 10 and 4, which correspond to 10 and 25% of the data being noise, respectively. Table 7.1 contains the comparison in velocity magnitude and angles in comparison with the ground truth. For a visual comparison, see Table 7.2. For an SNR of 4, the result starts to deviate noticeably, the streamlines have a slightly different trajectory. However, the method still provides a result that is close to the reference flow.

Besides a qualitative visualization of the flow using streamlines, derived flow features can be computed. One such feature is the λ_2-criterion, as explained in Chapter 5.3, that is commonly used to locate and visualize vortex cores and was computed for both the measured data as well as the data after our method was applied. As shown by Figure 7.6, the resulting vortex cores present in the after our method was applied match the measured data closely, indicating that features are preserved.

Comparison with previous work

We compare with previous methods for denoising PC-MRI data by computing the remaining divergence and difference with the input as shown in Figure 7.8. In the figure, the same slice shown for multiple approaches to illustrate the remaining divergence and the difference between the velocity vector angles and magnitudes with the measured data. All methods are close to the measured data, except one, however, the amount of divergence that remains present is minimul using our method. Ideally, the differences with the measured data should be small and the amount of divergence that that is present in the data should be zero. Since the measured data is not divergence-free, a compromise is needed, e.g., to have a smaller amount of divergence the difference with the measured data must be increased. However, for voxels where the data is less reliable more changes can be expected. The reliability of the used data is shown in Figure 7.7. To this end we compare our method with existing denoising methods that have as goal to remove

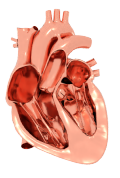
the divergence, namely *Divergence-Free Wavelets* (DFW) (with and without cycling spinning) [100], *Finite Difference Method* (FDM) [94] and *Radial Basis Functions* (RBF) [98]. Furthermore, we compare the result with the approach from Chapter 6, denoted by de Hoon et al. To evaluate the result, the remaining divergence and difference with the input measurement data is plotted. Note that the difference is not weighted with the reliability for a direct comparison. The difference with the input data is assessed by the absolute difference in velocity magnitude and angle between the velocity vectors. This comparison is shown in Figure 7.8. Besides the approach described previously in Chapter 6, which suffers from damping, all methods have small differences in velocity magnitude and angle. However, none of the methods we compare with is able to remove all divergence which corresponds with the findings of Sereno et al. [37]. Note that most divergence is present where the reliability is relatively low, i.e., near the vessel wall. Additionally, since the existing denoising approaches did not remove much of the divergence, we apply two possible approaches, i.e., a denoising method D that focuses on the reduction of noise in the data, and our pressure solver P, that targets the reduction of divergence. This should circumvent the bias of the approaches towards a specific aspect of the measured flow. For this we chose to use DFW with spin, which, of the tested denoising approaches, yields results that are the closets to the measured data. The result is shown in Figure 7.8, under the label $P(D)$. Note that the result is comparable to applying either of the approaches individually. The amount of divergence that remains present in the data is lowest for our method, suggesting that our method is able to remove more divergence. Further note that, when $\beta = 0$ for our method, i.e., the divergence term is neglected, the result is closer to the measured data as is to be expected, however, more divergence remains present.

7.5.4. Interpolation

A common approach to test any interpolation method is to remove parts of the data and see how well the interpolation can reconstruct the removed information. To evaluate the *spatial interpolation* capability of our method, all voxels for which one of its coordinates is even are removed from the data. Therefore, the downsampled data is 1/8 of the original size, see Table 7.2.

We apply our spatial interpolation scheme from Section 7.4.2 on this lower resolution data and compare the result with the input data in Table 7.1. Visually, the resulting flow field deviates slightly from the expected, left-out velocity field mainly due the loss of small-scale details, as can be seen in Table 7.2.

For the evaluation of the *temporal interpolation* described in Section 7.4.3 the same concept is applied. Therefore, one of the measured phases is removed, and the duration of the phases is doubled. The final row in Table 7.2 shows the input field, which is the linearly interpolated in between two consecutive phases. Linear interpolation is often used for the visualization of PC-MRI data, however, it does not guarantee a field that is physically-plausible. Visually, the result of our method matches the reference flow data well and better than linear interpolation. This is supported by the small differences in velocity magnitude and angle differences, as given in Table 7.1. This is also shown by Figure 7.9 that gives a direct comparison



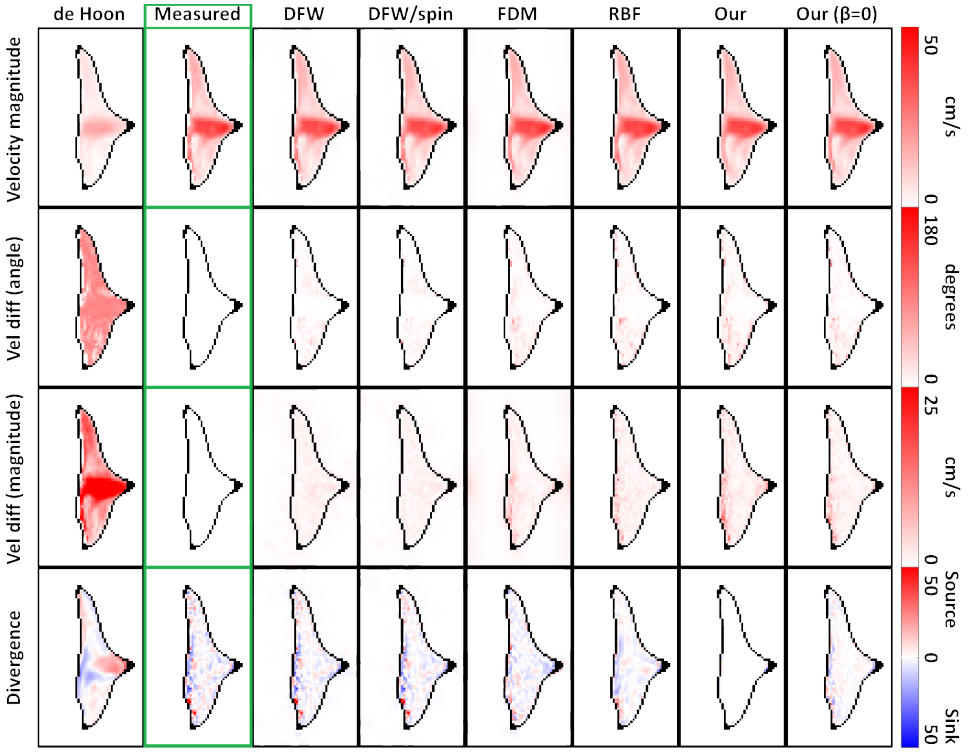


Figure 7.8: A visual comparison of our method with previous methods; the input measurement is highlighted by the green border. The same slice is shown in all images. The top row shows the velocity magnitude of the resulting field, while the second and third row show the difference with the measured input data. The bottom row shows the divergence that is present in the resulting velocity field. Note that all methods, except for our previous method from Chapter 6 (denoted by "de Hoon"), are close to the measured data, as shown by the difference between the velocity vector angles and magnitudes. The divergence that remains present in the data is minimal using our method. Note that our previous method is shown separately to allow for a more direct visual comparison of the remaining methods.

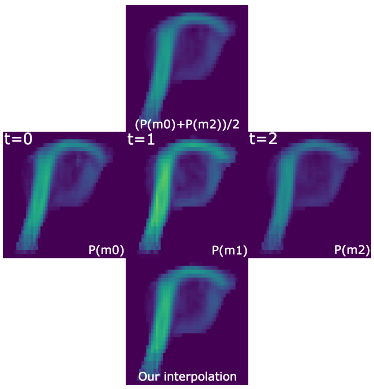


Figure 7.9: Comparison of temporal interpolation based on two pressure solved velocity fields (left and right) using linear interpolation (top), the expected pressure solved measured field (center) and our approach (bottom).

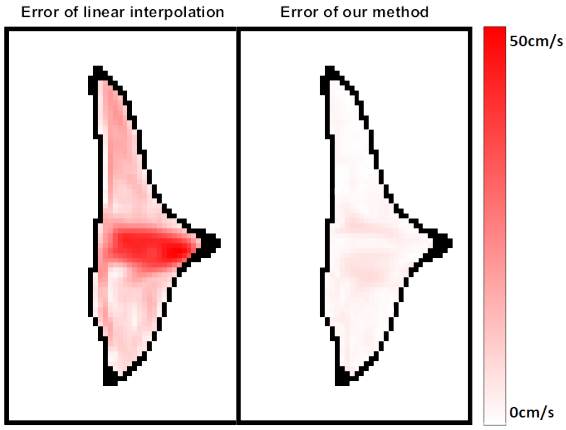


Figure 7.10: The error in comparison to a single time step of the Navier-Stokes simulation and linear interpolation (left) and our approach (right). Our approach has a small error. Linear interpolation on the other hand has a large error in the range of the velocity magnitude of the underlying velocity field.

with linear interpolation, the reference measured data and our method. Moreover, we compare our method and linear interpolation with a single time step of the Navier-Stokes simulation without minimization. That is, we evaluate our approach to determine whether it fulfils the Navier-Stokes equation by comparing our result with the right-hand side of the Navier-Stokes equation given by Equation 2.5 in Chapter 2.3. The right-hand side this equation provides us with an updated velocity field after a single time step of the Navier-Stokes simulation. The changes in the velocity field after this time step provides us with the left-hand side of the equation, i.e., the temporal derivative $\frac{\partial \vec{u}}{\partial t}$. Note that this temporal derivative can also be calculated from any two consecutive velocity fields by taking the difference between the two velocity fields: $\frac{\partial \vec{x}}{\partial t} = \vec{x}_{t+1} - \vec{x}_t$. Hence, we can compare the temporal derivative of different approaches to the solution provided by the Navier-Stokes simulation and compute the error: $error = \frac{\partial \vec{u}}{\partial t} - \frac{\partial \vec{x}}{\partial t}$. Figure 7.10 shows the magnitude of this error for both linear interpolation and for our approach. It shows that our approach has only a small error. Note that, linear interpolation and higher order interpolation methods do not reproduce the influence of advection that is present in flow data, which could explain the large error.

7.5.5. Vorticity near the aortic valve

Next, we demonstrate the advantage of our method for not requiring an accurate model of the anatomy, which in some cases, can be very difficult or impossible to model. This underlying anatomy can have a big impact on the resulting flow, for example, the flow near the heart valves. For example, in the aorta valve, three

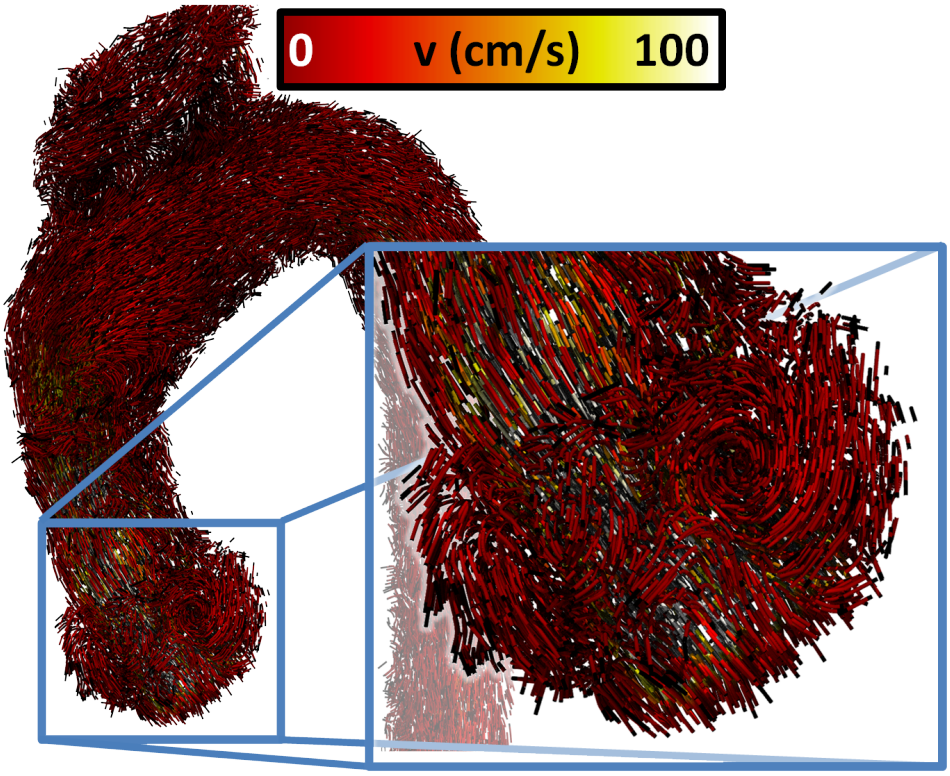


Figure 7.11: Measured flow in the aorta: vortices can be seen in the aorta root as indicated by the blue box.

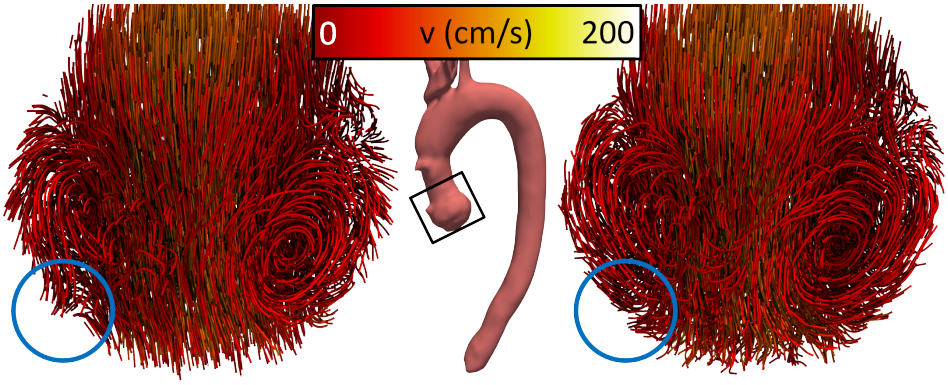


Figure 7.12: Flow in the aorta root showing the vorticity in the sinuses. The data is spatially interpolated using our method. Note that the vortices that help closing the valve leaflets are maintained. Short streamlines are used seeded from a spherical seeding region.

sinuses exist that correspond to the three cusps of the aortic valve. These sinuses are known to be important for an efficient blood flow. The vortices that form in these sinuses help to close the valve leaflets [5, 14, 19] and prevent regurgitation of the blood. This interaction between flow and anatomy is difficult to model [28] due to the small scale and lack of a possibility to measure this phenomena in-vivo [173].

The vortices, however, can be measured using PC-MRI, as can be seen in Figure 7.11. This data has a resolution of $127\times127\times23$ with a voxel size of $2.5\times2.5\times2.5$ millimetres. Since our method targets the whole velocity field, and not just in and outflow conditions, it can correct the (measured) flow in the sinuses without removing the important vortices present in the PC-MRI data. Even without a model of the aortic valve and its leaflets, our method maintains the vortices. Figure 7.12 shows the result of applying our denoising and spatial interpolation on the flow near the aortic valve.

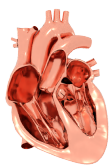
7.5.6. Circle of Willis

The circle of Willis is a relatively small circular vessel structure that supplies blood to the brain and the surrounding structures. The arteries of the circle of Willis have diameters smaller than 1 centimetres. Although the flow is measured using PC-MRI [189], the SNR is negatively affected, as a very fine spatial resolution is needed. The data used in this experiment has a resolution of $383 \times 383 \times 39$, with a voxel size of $0.47 \times 0.47 \times 0.5$ millimetres. Figure 7.13 shows the measured flow in a section of the circle of Willis. Note the artefacts and noise present in this data. For example, there is flow perpendicular to the main flow direction near a vortex, which is highly unlikely. Our method does correct for this inaccuracy and preserves nearby flow details, such as vortices, that are present in the data, as can be seen in Figure 7.14.

7.6. Performance

For most of the evaluations using the phantom data and the upsampled aorta root flow, our method needed between 200 and 300 iterations to converge and takes around 5 to 10 minutes. At most 8GB of memory was used. Temporal interpolation required much more memory (around 20GB) and takes around half an hour to converge. The circle of Willis data required more iterations to converge (around 600), which in the worst case took 45 minutes to converge. Note that this data has a relatively high resolution and low SNR. The experiments were performed on a Desktop computer with an Intel i7-3820 CPU and 32GB of memory.

Note that all evaluations could be executed in parallel on multiple machines or a computer cluster. Furthermore, parallelization of the code might be possible, but further complicates the computation of the gradient of the cost functionals.



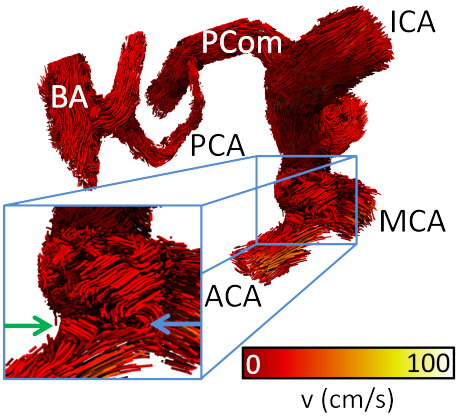


Figure 7.13: Streamline visualization of the measured flow in a section of the circle of Willis. Note the unlikely flow perpendicular to the main flow direction and the nearby vortex indicated by the green and blue arrows, respectively. The arteries shown are the basilar artery (BA), posterior cerebral artery (PCA), posterior communicating artery (PCom), internal carotid artery (ICA), middle cerebral artery (MCA) and anterior cerebral artery (ACA).

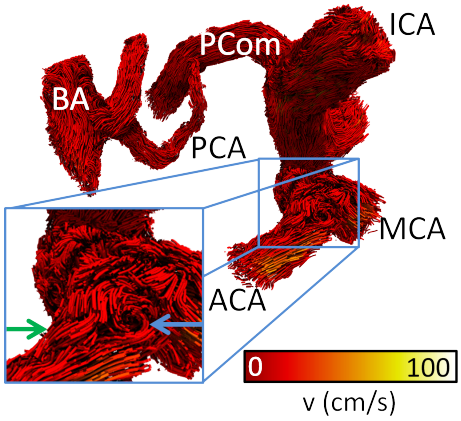


Figure 7.14: Streamline visualization of the corrected flow in a section of the circle of Willis. Our method corrected the perpendicular flow, as shown by the green arrow, while the nearby vortex is maintained (blue arrow).

7.7. Conclusion and Future work

4D PC-MRI provides measurements of time-varying volumetric blood flows. Despite providing the most complete data amongst all imaging methods, it is still prone to artefacts, noise and has limited spatial and temporal resolutions. Hence, the derived flow does not follow the fluid physics laws.

In this Chapter, we presented a novel data-assimilation approach for PC-MRI that combines a physics-based model and measured PC-MRI data to obtain physically-plausible, patient-specific data. Our methodology minimizes the error with respect to the measured data and has a limited sensitivity with respect to the accuracy of the given anatomical model.

We have shown, using phantom data, that the resulting method is capable of denoising noisy PC-MRI measurements and yields physically-plausible data, with minimal changes to the measured flow field. Furthermore, we have shown that our method enables physics-based interpolation in both the spatial and temporal domains. In addition to using phantom data, we have shown, as a proof of concept, that our method preserves important features in PC-MRI flow data. However, more extensive validation and comparison will be needed for the adaptation of the method to specific applications. Such validation is tedious and requires effort beyond out of the scope of this work.

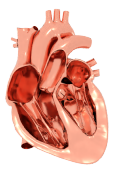
Our approach does not specifically avoid potential local minima. However, since the minimization is initialized close to the measured data, a local minimum would be

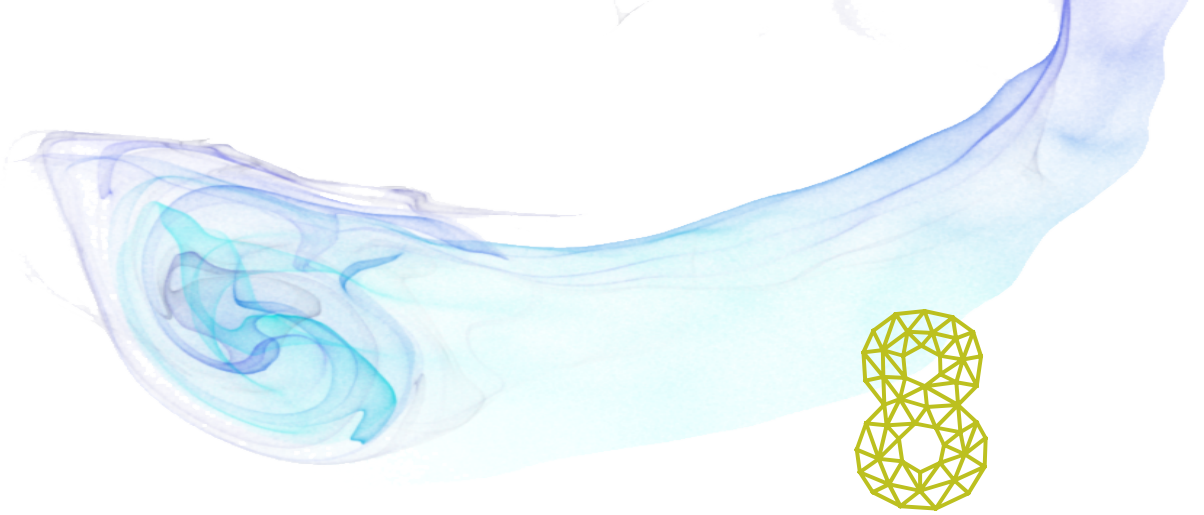
relatively close to the measured, target, data. To avoid local minima other optimization methods could be tested, such as simulated annealing, introducing randomness in the initialization of the minimization process and therefore using multiple starting points for the minimization. In this case, finding the optimal solution would be more likely. However, preliminary experiments seem to yield velocity fields that are close to the current solution, moreover, the computation time naturally increases. Nevertheless, it would be insightful to further investigate local-minima avoidance.

Another improvement would be the support of soft boundaries, i.e., allowing some flow to exist outside the segmentation. In this way, we can reduce the sensitivity of our method to the given anatomy. This could potentially be achieved by penalizing flows outside the segmentation. Another option would be to use control forces, for example using the geometric potential field from Hong et al. [111].

In the future, our approach could be used to reconstruct compressed PC-MRI data or enhance measured data. For the measured data, the spatial and temporal resolutions determine the signal-to-noise ratio. It would be interesting to determine scanning parameters for which we can achieve the highest noiseless data after applying our spatial and temporal denoising. On the longer term, the approach could potentially be used to provide prognostic information for treatment planning. Influence on the hemodynamics can be inspected for different morphological changes, or for different intervention schemes, such as stent or prosthetic valve placement. Ideally, the processing of the data should be performed on the fly.

To the extent of our knowledge, we presented the first data assimilation method for denoising and temporal interpolation of PC-MRI in Chapter 6. In this Chapter we extended that work and introduced an optimization method that allows for more flexibility and control and yields results that are closer to the measured data while maintaining important flow features. Moreover, the extended method provides denoising and both supports spatial & temporal interpolation of the data.





8 Conclusion

Cardiovascular Diseases (CVDs) have been the main cause of dead worldwide for many years [6–8]. As such, gaining more knowledge of the occurrence and progression of these diseases is paramount. One aspect of CVDs that plays an important role [1–3] is the blood flow within the cardiovascular system. This blood flow can be measured using, for example, *Phase-Contrast Magnetic Resonance Imaging* (PC-MRI). PC-MRI measurements surpass all other imaging methods in quality and completeness for measuring time-varying volumetric blood-flow data. Furthermore, this data have been shown to have the potential to improve both diagnosis and risk assessment of CVDs. The data consists of 3D vector field data that changes over time, i.e., unsteady flow. 3D unsteady flow analysis and visualization is a challenging and unsolved research problem. Visual analysis is relevant to understand the information provided by this rather novel PC-MRI modality. The quality of the measured PC-MRI data poses extra challenges to the flow visualization approaches.

In this thesis, we focussed on advancing flow visualization methods dealing with specific aspects of blood-flow visualization while considering the limitations of the acquisition method. To provide tools that aid a fast and flexible exploration of the blood-flow data, we proposed two novel visualization approaches. In Chapter 4, we introduced streak visualizations for blood-flow visualization that allow the visualization of time-dependent flow features, e.g., the progression of a vortex through time. Our collaborators adapted this framework to identify and visualize the vortices that are present in the aorta root. Moreover, the visualization of the impact of uncertainty due to measurement noise on the resulting visualization provided the users with new means to evaluate the reliability.

Due to the high variability of the flow in the heart and the constantly varying anatomy, additional scans and labour-intensive registration are commonly necessary. Therefore, a specialized framework was introduced in Chapter 5 for the visualization of the blood flow in the heart. The introduced framework provides a way to quickly explore the cardiac flow using only the 4D data from a single PC-MRI scan and no additional imaging. Furthermore, the framework minimizes the work

required for registration by introducing half ellipsoids to estimate the shape of the heart which can be placed using three clicks.

Despite flexible and fast exploration methods, the data itself is prone to noise, artefacts, and has a rather limited resolution, both temporally as well as spatially, like any measurement of physical phenomena. Therefore, PC-MRI data itself does not fulfil physical fluid laws. For data analysis and visualization, physically-plausible and high-resolution data is necessary to achieve reliable conclusions. Additionally, it is often difficult to distinguish noise and artefacts from important flow features. *Computational Fluid Dynamics* (CFD), on the other hand, provides means to generate high-resolution physically-plausible flow. However, flow simulations are highly dependent on the underlying anatomy and boundary conditions, which can be difficult or impossible to model adequately with current techniques, and, furthermore, assumptions needed for the simulation model are not always sufficiently known. Moreover, the spatial and temporal resolutions determine the signal-to-noise ratio, therefore, it would be interesting to determine scanning parameters for which we can achieve the highest resolution and noiseless data beforehand. This could potentially shorten and simplify the acquisition process. In this thesis we proposed two novel approaches that improve the quality of the measured data by combining the measured data with simulation models. The main goal of these methods was to keep the best of both worlds, i.e., keep the patient specific characteristics of the measurements while making the flow physically plausible using the simulation models, and, therefore, closer to the real flow. Chapter 6 introduced a method to increase the temporal resolution and reduce the amount of noise in the data using a simulation model. The simulation model used is initialized and steered by the measured patient data. In Chapter 7 a data assimilation technique was presented to remove the noise and artefacts as well as generate physically-plausible flow close to the measured data. Furthermore, this method also allowed us to increase the spatial and temporal resolution and improve the visualization of the blood flow.

On the longer term, such approaches can provide prognostic information for treatment planning. The influence on the hemodynamics could be inspected before treatment to determine the change in flow due to different morphological changes, or for different intervention schemes, such as stent or prosthetic valve placement. Since our approaches have a relatively low computation time this would allow for a relatively fast inspection of multiple options. Currently, however, the code is not yet optimized for performance and, ideally, the computations should be performed on the fly.

The methods proposed in this thesis have been validated to a limited extend. Validation of PC-MRI and specifically of the techniques proposed in this dissertation is very challenging due to the lack of a ground truth. There is no analytically physical flow that is complex enough to be considered for validation. Often validation based on synthetic phantoms can be used as a substitute for an analytical ground truth, since they can be measured for a longer time and under controlled circumstances. While phantoms provide a means to evaluate and compare to simulations, they do not provide real patient data. Therefore, an in-depth clinical study based a high number data sets analysed by multiple experts would provide a more complete

evaluation of the presented methods. Such an evaluation was considered outside the scope of this thesis given the complexity of achieving such a study. However, we hope that the proposed techniques can provide a bases for further analysis.

Finally, the work presented in this thesis, provides novel visualization and data assimilation approaches for PC-MRI data that aim to improve the current visualization and analysis approaches. Despite the future challenges, we believe the presented methods provide a basis for new research and to finally improve the analysis and understanding of the hemodynamics and cardiovascular diseases. Still a lot of work is needed to be able to get a complete and robust assimilation model. The assumptions made regarding, e.g., no viscosity and stiffness of the vessel walls, are not applicable to all situations and further analysis is needed. Hence, this work can be seen as a starting point for future work.

GLOSSARY

Abbreviations

3D-VAR 3-Dimensional Variational Assimilation

CFD Computational Fluid Dynamics.
A field of science that deals with analyses and solving problems that involve fluid flows using the computer.

CFL Courant-Friedrichs-Lewy

CPU Central Processing Unit

CT Computed Tomography.
A medical imaging approach.

CVDs Cardiovascular Diseases.
The CVDs relevant for this thesis are discussed in Section 2.1.4.

DFW Divergence-Free Wavelets.
A divergence filter using divergence-free wavelets.

FA Flip Angle

FDM Finite Difference Method

FEM Finite Element Method

FLIP Fluid Implicit Particle.
An approach that can be used for the simulation of fluids.

GPU Graphics Processing Unit

L-BFGS Limited-memory Broyden-Fletcher-Goldfarb-Shanno.
A quasi-Newton type minimization algorithm.

LCC Left Coronary Cusp

lerp linear interpolation

LIC Line Integral Convolution.
Flow visualization technique using noise textures.

LSM Lit Sphere Map

LVE Left Ventricle Ellipsoid

MAC Marker-And-Cell.
A grid-based spatial discretization for the simulation of, for example, fluids.

MRI Magnetic Resonance Imaging.
A medical imaging approach, acquisition is explained in Section 2.2.2.

NCC Non-Coronary Cusp

ODE Ordinary Differential Equation

PC-MRI	Phase-Contrast Magnetic Resonance Imaging. The PC-MRI data is explained in Section 2.2.1 and 2.2.3.
PCA-M	Phase-Contrast Angiography Magnitude
PCA-P	Phase-Contrast Angiography Phase
PDE	Partial Differential Equation
PIC	Particle In Cell. An approach that can be used for the simulation of fluids.
RBF	Radial Basis Functions. A divergence filter using convolution and divergence-free radial basis functions.
RCC	Right Coronary Cusp
RVE	Right Ventricle Ellipsoid
SNR	Signal-to-Noise Ratio
SPH	Smoothed Particle Hydrodynamics. An approach that can be used for the simulation of fluids.
T_E	Echo Time
TMIP	Temporal Maximum Intensity Projection. When multiple measurements over time exists, the TMIP can be computed. It stores per voxel the maximum value of that voxel had for all measurements.
T_R	Repetition Time
v_{enc}	velocity encoding speed
VSRR	Valve Sparing Aortic Root Replacement
WSS	Wall Shear Stress

BIBLIOGRAPHY

References

[1] L. Hiratzka, G. Bakris, J. Beckman, R. Bersin, V. Carr, D. Casey Jr, K. Eagle, L. Hermann, E. Isselbacher, E. Kazerooni, N. Kouchoukos, B. Lytle, D. Milewicz, D. Reich, S. Sen, J. Shinn, L. Svensson, and D. Williams, *Guidelines for the diagnosis and management of patients with thoracic aortic disease,* Journal of the American College of Cardiology **55**, e27 (2010).

[2] M. Markl, A. Frydrychowicz, S. Kozerke, M. Hope, and O. Wieben, *4D flow MRI,* Journal of Magnetic Resonance Imaging **36**, 1015 (2012).

[3] M. Markl, S. Schnell, C. Wu, E. Bollache, K. Jarvis, A. Barker, J. Robinson, and C. Rigsby, *Advanced flow MRI: emerging techniques and applications,* Clinical Radiology **71**, 779 (2016), special Issue: Spotlight on Cardiovascular Imaging.

[4] K. D. Keele, *Leonardo da vinci, and the movement of the heart.* Proceedings of the Royal Society of Medicine **44**, 209 (1951).

[5] M. M. Bissell, E. Dall'Armellina, and R. P. Choudhury, *Flow vortices in the aortic root: in vivo 4D-MRI confirms predictions of leonardo da vinci,* European Heart Journal **35**, 1344 (2014).

[6] A. S. Go, D. Mozaffarian, V. L. Roger, E. J. Benjamin, J. D. Berry, W. B. Borden, D. M. Bravata, S. Dai, E. S. Ford, C. S. Fox, S. Franco, H. J. Fullerton, C. Gillespie, S. M. Hailpern, J. A. Heit, V. J. Howard, M. D. Huffman, B. M. Kissela, S. J. Kittner, D. T. Lackland, J. H. Lichtman, L. D. Lisabeth, D. Magid, G. M. Marcus, A. Marelli, D. B. Matchar, D. K. McGuire, E. R. Mohler, C. S. Moy, M. E. Mussolino, G. Nichol, N. P. Paynter, P. J. Schreiner, P. D. Sorlie, T. N. T. J. Stein, S. S. Virani, N. D. Wong, D. Woo, and M. B. Turner, *Heart disease and stroke statistics - 2013 update,* Circulation **127**, e6 (2013).

[7] D. Mozaffarian, E. Benjamin, A. Go, D. Arnett, M. Glaha, M. Cushman, S. de Ferranti, J. Després, H. Fullerton, V. Howard, M. Muffman, S. Judd, B. Kissela, D. Lackland, J. Lichtman, L. Lisabeth, S. Liu, R. Mackey, D. Matchar, D. McGuire, E. M. 3rd, C. Moy, P. Muntner, M. Mussolino, K. Nasir, R. Neumar, G. Nichol, L. Palaniappan, D. Pandey, M. Reeves, C. Rodriquez, P. Sorlie, J. Stein, A. Towfighi, T. Turan, S. Virani, J. Willey, D. Woo, R. Yeh, and M. Turner, *Heart disease and stroke statistics–2015 update: a report from the American Heart Association.* Circulation **131**, e29 (2015).

[8] E. J. Benjamin, S. S. Virani, C. W. Callaway, A. M. Chamberlain, A. R. Chang, S. Cheng, S. E. Chiuve, M. Cushman, F. N. Delling, R. Deo, S. D. D. Ferranti, J. F. Ferguson, M. Fornage, C. Gillespie, C. R. Isasi, M. C. Jiménez, L. C. Jordan, S. E. Judd, D. Lackland, J. H. Lichtman, L. Lisabeth, S. Liu, C. T. Longenecker, P. L. Lutsey, J. S. Mackey, D. B. Matchar, K. Matsushita, M. E. Mussolino, K. Nasir, M. O'flaherty, L. P. Palaniappan, A. Pandey, D. K. Pandey, M. J. Reeves, M. D. Ritchey, C. J. Rodriguez, G. A. Roth, W. D. Rosamond, U. K. Sampson, G. M. Satou, S. H. Shah, N. L. Spartano, D. L. Tirschwell, C. W. Tsao, J. H. Voeks, J. Z. Willey, J. T. Wilkins, J. H. Wu, H. M. Alger, S. S. Wong, and P. Muntner, *Heart disease and stroke statistics–2018 update: A report from the American Heart Association,* Circulation **137** (2018), 10.1161/CIR.0000000000000558.

[9] P. R. Moran, *A flow velocity zeugmatographic interlace for nmr imaging in humans,* Magnetic Resonance Imaging **1**, 197 (1982), second Annual Meeting of the Society for Magnetic Resonance Imaging.

[10] L. Wigström, L. Sjöqvist, and B. Wranne, *Temporally resolved 3d phase-contrast imaging,* Magnetic Resonance in Medicine **36**, 800 (1996).

[11] P. Kilner, G. Yang, R. Mohiaddin, D. Firmin, and D. Longmore, *Helical and retrograde secondary flow patterns in the aortic arch studied by three-directional magnetic resonance velocity mapping.* Circulation **88**, 2235 (1993).

[12] S. Katayama, N. Umetani, S. Sugiura, and T. Hisada, *The sinus of valsalva relieves abnormal stress on aortic valve leaflets by facilitating smooth closure.* The Journal of thoracic and cardiovascular surgery **136**, 1528 (2008).

[13] K. J. Grande-Allen, R. P. Cochran, P. G. Reinhall, and K. S. Kunzelman, *Recreation of sinuses is important for sparing the aortic valve: a finite element study.* The Journal of thoracic and cardiovascular surgery **119**, 753 (2000).

[14] B. Bellhouse and F. Bellhouse, *Mechanism of closure of the aortic valve,* Nature (1968), 10.1038/217086b0.

[15] G. Pisani, R. Scaffa, O. Ieropoli, E. M. Dell'Amico, D. Maselli, U. Morbiducci, and R. De Paulis, *Role of the sinuses of valsalva on the opening of the aortic valve.* The Journal of thoracic and cardiovascular surgery **145**, 999 (2013).

[16] A. Birkeland, D. M. Ulvang, K. Nylund, T. Hausken, O. H. Gilja, and I. Viola, *Doppler-based 3d blood flow imaging and visualization,* in *Proceedings of the 29th Spring Conference on Computer Graphics*, SCCG '13 (ACM, New York, NY, USA, 2013) pp. 115–122.

[17] T. A. Hope and R. J. Herfkens, *Imaging of the thoracic aorta with time-resolved three-dimensional Phase-Contrast MRI: A review,* Seminars in Thoracic and Cardiovascular Surgery **20**, 358 (2008).

[18] M. Markl, M. T. Draney, D. C. Miller, J. M. Levin, E. E. Williamson, N. J. Pelc, D. H. Liang, and R. J. Herfkens, *Time-resolved three-dimensional magnetic resonance velocity mapping of aortic flow in healthy volunteers and patients after valve-sparing aortic root replacement,* The Journal of Thoracic and Cardiovascular Surgery **130**, 456 (2005).

[19] J. Vendrik, E. S. Farag, N. H. L. C. de Hoon, J. Kluin, and J. Baan, *Presence of aortic root vortex formation after TAVI with CENTERA confirmed using 4D-flow magnetic resonance imaging,* The International Journal of Cardiovascular Imaging (2018), 10.1007/s10554-018-1413-2.

[20] A. Berger, *Magnetic resonance imaging,* BMJ (Clinical research ed.) **324**, 35 (2002).

[21] B. Köhler, S. Born, R. F. P. van Pelt, A. Hennemuth, U. Preim, and B. Preim, *A survey of cardiac 4D PC-MRI data processing,* Computer Graphics Forum **36**, 5 (2016).

[22] J. Lotz, C. Meier, A. Leppert, and M. Galanski, *Cardiovascular flow measurement with phase-contrast mr imaging: Basic facts and implementation,* RadioGraphics **22**, 651 (2002), pMID: 12006694, https://doi.org/10.1148/radiographics.22.3.g02ma11651 .

[23] R. Gasteiger, *Visual Exploration of Cardiovascular Hemodynamics*, Ph.D. thesis, Otto-von-Guericke University Magdeburg (2014).

[24] O. Friman, A. Hennemuth, A. Harloff, J. Bock, M. Markl, and H. Peitgen, *Probabilistic 4d blood flow mapping,* MICCAI **13** (2010), 10.1007/978-3-642-15711-0_52.

[25] O. Friman, A. Hennemuth, A. Harloff, J. Bock, M. Markl, and H. Peitgen, *Probabilistic 4D blood flow tracking and uncertainty estimation,* Medical Image Analysis **15**, 720 (2011), special Issue on the 2010 Conference on Medical Image Computing and Computer-Assisted Intervention.

[26] H. Gudbjartsson and S. Patz, *The Rician distribution of noisy MRI data,* Magnetic Resonance in Medicine **34**, 910 (1995).

[27] J. Stam, *Stable fluids,* in *SIGGRAPH Proceedings of the Conference on Computer Graphics and Interactive Techniques* (1999) pp. 121–128.

[28] P. D. Morris, A. Narracott, H. von Tengg-Kobligk, D. A. Silva Soto, S. Hsiao, A. Lungu, P. Evans, N. W. Bressloff, P. V. Lawford, D. R. Hose, and J. P. Gunn, *Computational fluid dynamics modelling in cardiovascular medicine,* Heart (2015), 10.1136/heartjnl-2015-308044, https://heart.bmj.com/content/early/2015/10/28/heartjnl-2015-308044.full.pdf .

[29] M. Müller, D. Charypar, and M. Gross, *Particle-based fluid simulations for interactive applications,* in *ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2003) pp. 154–159.

[30] A. Selle, N. Rasmussen, and R. Fedkiw, *A vortex particle method for smoke, water and explosions,* ACM Transactions on Graphics **24**, 910 (2005).

[31] C. A. Taylor and C. A. Figueroa, *Patient-specific modeling of cardiovascular mechanics,* Annual Review of Biomedical Engineering **11**, 109 (2009).

[32] J. Brackbill, D. Kothe, and H. Ruppel, *FLIP: A low-dissipation, particle-in-cell method for fluid flow,* Computer Physics Communications **48**, 25 (1988).

[33] Y. Zhu and R. Bridson, *Animating sand as a fluid,* ACM Transactions on Graphics **24**, 965 (2005).

[34] C. Batty, F. Bertails, and R. Bridson, *A fast variational framework for accurate solid-fluid coupling,* ACM Transactions on Graphics **26**, 100:1 (2007).

[35] C. Batty and R. Bridson, *Accurate viscous free surfaces for buckling, coiling, and rotating liquids,* in *ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2008) pp. 219–228.

[36] R. Bridson, *Fluid Simulation for Computer Graphics* (A.K. Peters, 2008).

[37] M. F. Sereno, B. Köhler, and B. Preim, *Comparison of divergence-free filters for cardiac 4D PC-MRI data,* in *Bildverarbeitung für die Medizin 2018*, edited by A. Maier, T. M. Deserno, H. Handels, K. H. Maier-Hein, C. Palm, and T. Tolxdorff (Springer Berlin Heidelberg, Berlin, Heidelberg, 2018) pp. 139–144.

[38] R. van der Geest and P. Garg, *Advanced analysis techniques for intra-cardiac flow evaluation from 4D flow MRI,* Current Radiology Reports **4** (2016), AID - 10.1007/s40134-016-0167-7.

[39] R. van Pelt, J. O. Bescos, M. Breeuwer, R. E. Clough, M. E. Groller, B. ter Haar Romenij, and A. Vilanova, *Exploration of 4D MRI blood flow using stylistic visualization,* IEEE Transactions on Visualization and Computer Graphics **16**, 1339 (2010).

[40] S. Born, M. Markl, M. Gutberlet, and G. Scheuermann, *Illustrative visualization of cardiac and aortic blood flow from 4D MRI data,* in *IEEE Pacific Visualization* (2013) pp. 129 – 136.

[41] B. Köhler, R. Gasteiger, U. Preim, H. Theisel, M. Gutberlet, and B. Preim, *Semi-automatic vortex extraction in 4D PC-MRI cardiac blood flow data using line predicates,* IEEE Transactions on Visualization and Computer Graphics **19**, 2773 (2013).

[42] D. Lane, *Visualizing time-varying phenomena in numerical simulations of unsteady flows* (National Aeronautics and Space Administration, 1996).

[43] A. Vilanova, B. Preim, R. van Pelt, R. Gasteiger, M. Neugebauer, and T. Wischgoll, *Visual exploration of simulated and measured blood flow,* Computing Research Repository - arXiv **abs/1209.0999** (2012).

[44] J. van Wijk, *Rendering surface-particles,* in *Visualization, 1992. Visualization '92, Proceedings., IEEE Conference on* (1992) pp. 54–61.

[45] P. Kipfer, M. Segal, and R. Westermann, *Uberflow: A gpu-based particle engine,* in *Proceedings of the ACM SIGGRAPH/EUROGRAPHICS Conference on Graphics Hardware*, HWWS '04 (ACM, New York, NY, USA, 2004) pp. 115–122.

[46] W. von Funck, T. Weinkauf, H. Theisel, and H.-P. Seidel, *Smoke surfaces: An interactive flow visualization technique inspired by real-world flow experiments,* IEEE Trans. Vis. Comput. Graph. **14**, 1396 (2008).

[47] M. Falk and D. Weiskopf, *Output-sensitive 3d line integral convolution,* IEEE Transactions on Visualization and Computer Graphics **14**, 820 (2008).

[48] R. van Pelt, J. O. Bescos, M. Breeuwer, R. E. Clough, M. E. Groller, B. ter Haar Romenij, and A. Vilanova, *Interactive virtual probing of 4D MRI blood-flow,* IEEE Transactions on Visualization and Computer Graphics **17**, 2153 (2011).

[49] K. Lawonn, R. Gasteiger, and B. Preim, *Adaptive surface visualization of vessels with animated blood flow,* Computer Graphics Forum **33**, 16 (2014).

[50] F. Ferstl, K. Bürger, and R. Westermann, *Streamline variability plots for characterizing the uncertainty in vector field ensembles,* IEEE Transactions on Visualization and Computer Graphics **22**, 767 (2016).

[51] J. Jeong and F. Hussain, *On the identification of a vortex,* Journal of Fluid Mechanics **285**, 69 (1995).

[52] G. Haller, *An objective definition of a vortex,* Journal of Fluid Mechanics **525**, 1 (2005).

[53] R. van Pelt, A. Fuster, G. Claassen, and A. Vilanova, *Characterization of blood-flow patterns from phase-contrast MRI velocity fields,* in *EuroVis - Short Papers* (2014).

[54] F. Post, B. Vrolijk, H. Hauser, R. Laramee, and H. Doleisch, *The state of the art in flow visualisation: Feature extraction and tracking,* Computer Graphics Forum **22**, 775 (2003).

[55] C. Muelder and K. L. Ma, *Interactive feature extraction and tracking by utilizing region coherency,* in *2009 IEEE Pacific Visualization Symposium* (2009) pp. 17–24.

[56] D. Stalling, M. Zockler, and H. Hege, *Fast display of illuminated field lines,* IEEE Transactions on Visualization and Computer Graphics **3**, 118 (1997).

[57] H. Doleisch, M. Gasser, and H. Hauser, *Interactive feature specification for focus+context visualization of complex simulation data,* in *Proceedings of the Symposium on Data Visualisation 2003* (2003) pp. 239–248.

[58] R. van Pelt, S. Jacobs, B. ter Haar Romeny, and A. Vilanova, *Visualization of 4D blood-flow fields by spatiotemporal hierarchical clustering,* in *Computer Graphics Forum*, Vol. 31 (2012) pp. 1065–1074.

[59] M. Meuschke, S. Voß, B. Preim, and K. Lawonn, *Exploration of blood flow patterns in cerebral aneurysms during the cardiac cycle,* Computers & Graphics **72**, 12 (2018).

[60] M. Meuschke, S. Oeltze-Jafra, O. Beuing, B. Preim, and K. Lawonn, *Classification of blood flow patterns in cerebral aneurysms,* IEEE Transactions on Visualization and Computer Graphics **25**, 2404 (2019).

[61] G. Kindlmann, R. Whitaker, T. Tasdizen, and T. Moller, *Curvature-based transfer functions for direct volume rendering: Methods and applications,* in *Proceedings of the 14th IEEE Visualization 2003* (IEEE Computer Society, 2003).

[62] S. Bruckner and M. Gröller, *Style transfer functions for illustrative volume rendering,* Computer Graphics Forum **26**, 715 (2007).

[63] R. Gasteiger, M. Neugebauer, O. Beuing, and B. Preim, *The flowlens: A focus-and-context visualization approach for exploration of blood flow in cerebral aneurysms,* IEEE Transactions on Visualization and Computer Graphics **17**, 2183 (2011).

[64] M. Guttman, E. Zerhouni, and E. McVeigh, *Analysis of cardiac function from MR images,* IEEE computer graphics and applications **17**, 30 (1997).

[65] Z. Stankovic, B. Allen, J. Garcia, K. Jarvis, and M. Markl, *4D flow imaging with MRI,* Cardiovascular diagnosis and therapy **4**, 173 (2014).

[66] D. Weiskopf, K. Engel, and T. Ertl, *Volume clipping via per-fragment operations in texture-based volume visualization,* in *Proceedings of the Conference on Visualization '02* (IEEE Computer Society, 2002) pp. 93–100.

[67] A. Brambilla, R. Carnecky, R. Peikert, I. Viola, and H. Hauser, *Illustrative flow visualization: State of the art, trends and challenges,* in *Eurographics 2012 - State of the Art Reports* (The Eurographics Association, 2012).

[68] P. Sloan, W. Martin, A. Gooch, and B. Gooch, *The lit sphere: A model for capturing npr shading from art,* in *Proceedings of the Graphics Interface 2001 Conference, June 7-9 2001, Ottawa, Ontario, Canada* (2001) pp. 143–150.

[69] W. Sörgel and V. Vaerman, *Automatic heart localization from a 4D MRI dataset,* in *In Proc. of the SPIE Conference on Medical Imaging* (1997) pp. 333–344.

[70] J. Cousty, L. Najman, M. Couprie, S. Cl'ement-Guinaudeau, T. Goissen, and J. Garot, *Segmentation of 4D cardiac MRI: Automated method based on spatio-temporal watershed cuts,* Image and Vision Computing **28**, 1229 (2010).

[71] P. Yushkevich, J. Piven, H. Hazlett, R. Smith, S. Ho, J. Gee, and G. Gerig, *User-guided 3D active contour segmentation of anatomical structures: Significantly improved efficiency and reliability,* NeuroImage **31**, 1116 (2006).

[72] P. Kohlmann, S. Bruckner, A. Kanitsar, and M. Gröller, *Livesync: Deformed viewing spheres for knowledge-based navigation,* IEEE Transactions on Visualization and Computer Graphics **13**, 1544 (2007).

[73] H. Hu, Z. Gao, L. Liu, H. Liu, J. Gao, S. Xu, W. Li, and L. Huang, *Automatic segmentation of the left ventricle in cardiac MRI using local binary fitting model and dynamic programming techniques,* PLoS ONE **9**, 1 (2014).

[74] M. Lynch, O. Ghita, and P. Whelan, *Automatic segmentation of the left ventricle cavity and myocardium in MRI data,* Computers in Biology and Medicine **36**, 389 (2006).

[75] C. Cocosco, W. Niessen, T. Netsch, E. Vonken, G. Lund, A. Stork, and M. Viergever, *Automatic image-driven segmentation of the ventricles in cardiac cine MRI,* Journal of magnetic resonance imaging : JMRI **28**, 366 (2008).

[76] S. Kovalova, J. Necas, R. Cerbak, P. Malik, and J. Vespalec, *Echocardiographic volumetry of the right ventricle,* European journal of echocardiography **6**, 15 (2005).

[77] S. Djurcilova, K. Kima, P. Lermusiaux, and A. Pang, *Visualizing scalar volumetric data with uncertainty,* Computers and Graphics **26**, 239 (2002).

[78] T. Pang, M. Wittenbrink, and K. Lodha, *Approaches to uncertainty visualization,* The Visual Computer **13**, 370 (1997).

[79] J. Kniss, R. V. Uitert, A. Stephens, G. Li, T. Tasdizen, and C. Hansen, *Statistically quantitative volume visualization,* IEEE visualization , 287 (2005).

[80] C. Lundström, P. Ljung, A. Persson, and A. Ynnerman, *Uncertainty visualization in medical volume rendering using probabilistic animation,* Visualization and Computer Graphics **13**, 1648 (2007).

[81] C. Johnson and J. Huang, *Distribution-driven visualization of volume data,* IEEE Computer Graphics and Applications **15**, 734 (2009).

[82] S. K. Lodha, A. Pang, R. E. Sheehan, and C. M. Wittenbrink, *Uflow: visualizing uncertainty in fluid flow,* in *Visualization '96. Proceedings.* (1996) pp. 249–254.

[83] W. He, C. M. Chen, X. Liu, and H. W. Shen, *A bayesian approach for probabilistic streamline computation in uncertain flows,* in *2016 IEEE Pacific Visualization Symposium (PacificVis)* (2016) pp. 214–218.

[84] A. Pang, *Visualizing uncertainty in geo-spatial data,* in *Proceedings of the Workshop on the Intersections between Geospatial Information and Information Technology* (2001).

[85] M. Hlawatsch, P. Leube, N. Wolfgang, and D. Weiskopf, *Flow radar glyphs–static visualization of unsteady flow with uncertainty,* IEEE transactions on visualization and computer graphicsc **17**, 1949 (2011).

[86] R. Botchen, D. Weiskopf, and T. Ertl, *Texture-based visualization of uncertainty in flow fields,* in *VIS 05. IEEE Visualization, 2005.* (2005) pp. 647–654.

[87] M. Otto, T. Germer, H.-C. Hege, and H. Theisel, *Uncertain 2d vector field topology,* Computer Graphics Forum **29**, 347 (2010).

[88] H. Bhatia, S. Jadhav, P. T. Bremer, G. Chen, J. A. Levine, L. G. Nonato, and V. Pascucci, *Flow visualization with quantified spatial and temporal errors using edge maps,* IEEE Transactions on Visualization and Computer Graphics **18**, 1383 (2012).

[89] C. Petz, K. Pöthkow, and H. Hege, *Probabilistic local features in uncertain vector fields with spatial correlation,* Computer Graphics Forum **31**, 1045 (2012).

[90] M. Otto, T. Germer, and H. Theisel, *Uncertain topology of 3d vector fields,* in *2011 IEEE Pacific Visualization Symposium* (2011) pp. 67–74.

[91] H. Guo, W. He, T. Peterka, H. W. Shen, S. Collis, and J. Helmus, *Finite-time Lyapunov exponents and Lagrangian coherent structures in uncertain unsteady flows,* IEEE Transactions on Visualization and Computer Graphics **PP** (2016), 10.1109/TVCG.2016.2534560.

[92] M. Schwenke, A. Hennemuth, B. Fischer, and O. Friman, *A novel anisotropic fast marching method and its application to blood flow computation in Phase-Contrast MRI,* Methods of information in medicine **51** (2012), 10.3414/ME11.02.0032.

[93] B. Köhler, U. Preim, M. Grothoff, M. Gutberlet, K. Fischbach, and B. Preim, *Robust cardiac function assessment in 4d pc-mri data of the aorta and pulmonary artery,* Computer Graphics Forum **35**, 32 (2015).

[94] S. M. Song, S. Napel, G. H. Glover, and N. J. Pelc, *Noise reduction in three-dimensional phase-contrast MR velocity measurementsl,* Journal of Magnetic Resonance Imaging **3**, 587 (1993), https://onlinelibrary.wiley.com/doi/pdf/10.1002/jmri.1880030407 .

[95] M. H. Buonocore, *Algorithms for improving calculated streamlines in 3-d phase contrast angiography,* Magnetic Resonance in Medicine **31**, 22 (1994), https://onlinelibrary.wiley.com/doi/pdf/10.1002/mrm.1910310104 .

[96] N. Fatouraee and A. Amini, *Regularization of flow streamlines in multislice phase-contrast MR imaging,* IEEE Transactions on Medical Imaging **22**, 699 (2003).

[97] M. Loecher, S. Kecskemeti, P. Turski, and O. Wieben, *Comparison of divergence-free algorithms for 3D MRI with three-directional velocity encoding,* Journal of Cardiovascular Magnetic Resonance **14**, W64 (2012).

[98] J. Busch, D. Giese, L. Wissmann, and S. Kozerke, *Reconstruction of divergence-free velocity fields from cine 3D phase-contrast flow measurements,* Magnetic Resonance in Medicine **69**, 200 (2013), https://onlinelibrary.wiley.com/doi/pdf/10.1002/mrm.24221 .

[99] E. Bostan, S. Lefkimmiatis, O. Vardoulis, N. Stergiopulos, and M. Unser, *Improved variational denoising of flow fields with application to phase-contrast MRI data,* IEEE Signal Processing Letters **22**, 762 (2015).

[100] F. Ong, M. Uecker, U. Tariq, A. Hsiao, M. T. Alley, S. S. Vasanawala, and M. Lustig, *Robust 4D flow denoising using divergence-free wavelet transform,* Magnetic Resonance in Medicine **73**, 828 (2015).

[101] H. Bhatia, G. Norgard, V. Pascucci, and P. Bremer, *The Helmholtz-Hodge decomposition-a survey,* IEEE Transactions on Visualization and Computer Graphics **19**, 1386 (2013).

[102] T. B. Le, M. S. M. Elbaz, R. J. Van Der Geest, and F. Sotiropoulos, *High resolution simulation of diastolic left ventricular hemodynamics guided by four-dimensional flow magnetic resonance imaging data,* Flow, Turbulence and Combustion (2019), 10.1007/s10494-018-0003-7.

[103] M. Bocquet, H. Elbern, H. Eskes, M. Hirtl, R. Žabkar, G. Carmichael, J. Flemming, A. Inness, M. Pagowski, J. Pérez Camaño, P. Saide, R. S. Jose, M. Sofiev, J. Vira, A. Baklanov, C. Carnevale, G. Grell, and C. Seigneur, *Data assimilation in atmospheric chemistry models: current status and future prospects for coupled chemistry meteorology models,* Atmospheric Chemistry and Physics **15**, 5325 (2015).

[104] M. Rodell, P. Houser, U. Jambor, J. Gottschalck, K. Mitchell, C.-J. Meng, K. Arsenault, B. Cosgrove, J. Radakovich, M. Bosilovich, J. Entin, J. Walker, D. Lohmann, and D. Toll, *The global land data assimilation system,* Bulletin of the American Meteorological Society **85**, 381 (2004).

[105] U. Schneider, A. Becker, P. Finger, A. Meyer-Christoffer, M. Ziese, and B. Rudolf, *GPCC's new land surface precipitation climatology based on quality-controlled in situ data and its role in quantifying the global water cycle,* Theoretical and Applied Climatology **115**, 15 (2014).

[106] D. P. Dee, S. M. Uppala, A. J. Simmons, P. Berrisford, P. Poli, S. Kobayashi, U. Andrae, M. A. Balmaseda, G. Balsamo, P. Bauer, P. Bechtold, A. C. M. Beljaars, L. van de Berg, J. Bidlot, N. Bormann, C. Delsol, R. Dragani, M. Fuentes, A. J. Geer, L. Haimberger, S. B. Healy, H. Hersbach, E. V. Hólm, L. Isaksen, P. Kållberg, M. Köhler, M. Matricardi, A. P. McNally, B. M. Monge-Sanz, J.-J. Morcrette, B.-K. Park, C. Peubey, P. de Rosnay, C. Tavolato, J.-N. Thépaut, and F. Vitart, *The ERA-interim reanalysis: configuration and performance of the data assimilation system,* Quarterly Journal of the Royal Meteorological Society **137**, 553 (2011).

[107] A. C. Lorenc, N. E. Bowler, A. M. Clayton, S. R. Pring, and D. Fairbairn, *Comparison of hybrid-4DEnVar and hybrid-4DVar data assimilation methods for global NWP,* Monthly Weather Review **143**, 212 (2015).

[108] S. G. Penny, D. Behringer, J. Carton, E. Kalnay, and Y. Xue, *Towards an operational hybrid global ocean data assimilation system (hybrid-GODAS) at NCEP,* in *American Geophysical Union, Ocean Sciences Meeting 2016* (2016).

[109] N. Foster and D. Metaxas, *Controlling fluid animation,* in *Computer Graphics International* (1997) pp. 178–188.

[110] J. Gregson, I. Ihrke, N. Thürey, and W. Heidrich, *From capture to simulation - connecting forward and inverse problems in fluids,* (ACM Press, 2014) p. 10.

[111] J.-m. Hong and C.-h. Kim, *Controlling fluid animation with geometric potential,* Computer Animation and Virtual Worlds **15**, 147 (2004).

[112] Z. Pan and D. Manocha, *Efficient solver for spacetime control of smoke,* ACM Transactions on Graphics **36** (2017), 10.1145/3016963.

[113] N. Thürey, R. Keiser, M. Pauly, and U. Rüde, *Detail-preserving fluid control,* in *Proceedings of the 2006 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '06 (Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 2006) pp. 7–12.

[114] M. Chu and N. Thuerey, *Data-driven synthesis of smoke flows with CNN-based feature descriptors,* ACM Trans. Graph. **36**, 69:1 (2017).

[115] A. McNamara, A. Treuille, Z. Popović, and J. Stam, *Fluid control using the adjoint method,* ACM Trans. Graph. **23**, 449 (2004).

[116] M. B. Nielsen and B. B. Christensen, *Improved variational guiding of smoke animations,* Computer Graphics Forum **29**, 705 (2010), https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1467-8659.2009.01640.x .

[117] F. P. Glor, J. J. M. Westenberg, J. Vierendeels, M. Danilouchkine, and P. Verdonck, *Validation of the coupling of magnetic resonance imaging velocity measurements with computational fluid dynamics in a U bend.* Artificial Organs **26**, 622 (2002).

[118] N. B. Wood, S. J. Weston, P. J. Kilner, A. D. Gosman, and D. N. Firmin, *Combined MR imaging and CFD simulation of flow in the human descending aorta.* Journal of Magnetic Resonance Imaging **13**, 699 (2001).

[119] M. D'Elia, M. Perego, and A. Veneziani, *A variational data assimilation procedure for the incompressible Navier-Stokes equations in hemodynamics.* Journal of Scientific Computing **52**, 340 (2012).

[120] S. W. Funke, M. Nordaas, Ø. Evju, M. S. Alnæs, and K.-A. Mardal, *Variational data assimilation for transient blood flow simulations,* International Journal for Numerical Methods in Biomedical Engineering (2018).

[121] S. Ii, M. A. H. M. Adib, Y. Watanabe, and S. Wada, *Physically consistent data assimilation method based on feedback control for patient-specific blood flow analysis,* International Journal for Numerical Methods in Biomedical Engineering **34**, e2910 (2018), e2910 cnm.2910, https://onlinelibrary.wiley.com/doi/pdf/10.1002/cnm.2910 .

[122] T. Guerra, C. Catarino, T. Mestre, S. Santos, J. Tiago, and A. Sequeira, *A data assimilation approach for non-Newtonian blood flow simulations in 3D geometries,* Applied Mathematics and Computation **321**, 176 (2018).

[123] K. Funamoto, T. Hayase, A. Shirai, Y. Saijo, and T. Yambe, *Fundamental study of ultrasonic-measurement-integrated simulation of real blood flow in the aorta,* Annals of Biomedical Engineering **33**, 415 (2005).

[124] J. Heys, T. Manteuffel, S. McCormick, M. Milano, J. Westerdale, and M. Belohlavek, *Weighted least-squares finite elements based on particle imaging velocimetry data,* Journal of Computational Physics **229**, 107 (2010).

[125] P. Rajaraman, T. Manteuffel, M. Belohlavek, E. McMahon, and J. Heys, *Echocardiographic particle imaging velocimetry data assimilation with least square finite element methods,* Computers and Mathematics with Applications **68**, 1569 (2014).

[126] M. Alimohammadi, O. Agu, S. Balabani, and V. Díaz-Zuccarini, *Development of a patient-specific simulation tool to analyse aortic dissections: Assessment of mixed patient-specific flow and pressure boundary conditions,* Medical Engineering & Physics **36** (2014), 10.1016/j.medengphy.2013.11.003.

[127] V. Rispoli, J. Nielsen, K. Nayak, and J. Carvalho, *Computational fluid dynamics simulations of blood flow regularized by 3D phase contrast MRI,* BioMedical Engineering OnLine **14**, 1 (2015).

[128] M. F. Fathi, A. Bakhshinejad, A. Baghaie, D. Saloner, R. H. Sacho, V. L. Rayz, and R. M. D'Souza, *Denoising and spatial resolution enhancement of 4D flow MRI using proper orthogonal decomposition and lasso regularization,* Computerized Medical Imaging and Graphics **70**, 165 (2018).

[129] N. H. L. C. de Hoon, K. Lawonn, A. C. Jalba, E. Eisemann, and A. Vilanova, *InkVis: A High-Particle-Count Approach for Visualization of Phase-Contrast Magnetic Resonance Imaging Data,* in *Eurographics Workshop on Visual Computing for Biology and Medicine*, edited by B. Kozlíková, L. Linsen, P.-P. Vázquez, K. Lawonn, and R. G. Raidou (The Eurographics Association, 2019).

[130] J. Cebral, M. Castro, J. Burgess, R. Pergolizzi, M. Sheridan, and C. Putman, *Characterization of cerebral aneurysms for assessing risk of rupture by using patient-specific computational hemodynamics models,* American Journal of Neuroradiology **26**, 2550 (2005).

[131] P. M. Arvidsson, S. J. Kovacs, J. Töger, R. Borgquist, E. Heiberg, M. Carlsson, and H. Arheden, *Vortex ring behavior provides the epigenetic blueprint for the human heart,* in *Scientific reports*, Vol. 6 (2016).

[132] R. Carroll and H. Falsetti, *Retrograde coronary artery flow in aortic valve disease.* Circulation **54**, 494 (1976).

[133] T. McLoughlin, R. S. Laramee, R. Peikert, F. H. Post, and M. Chen, *Over two decades of integration-based, geometric flow visualization,* Computer Graphics Forum **29**, 1807 (2010).

[134] S. French, *Modelling, making inferences and making decisions: The roles of sensitivity analysis,* Top **11**, 229 (2003).

[135] E. Straszecka, *Combining uncertainty and imprecision in models of medical diagnosis,* Information Sciences **176**, 3026 (2006).

[136] J. von Spiczak, G. Crelier, D. Giese, S. Kozerke, D. Maintz, and A. Bunck, *Quantitative analysis of vortical blood flow in the thoracic aorta using 4D Phase Contrast MRI,* PLOS ONE **10**, 1 (2015).

[137] G. Bonneau, H. Hege, C. Johnson, M. Oliveira, K. Potter, P. Rheingans, and T. Schultz, *Scientific visualization: Uncertainty, multifield, biomedical, and scalable visualization,* (Springer London, 2014) Chap. Overview and State-of-the-Art of Uncertainty Visualization, pp. 3–27.

[138] K. Brodlie, R. Allendes Osorio, and A. Lopes, *A review of uncertainty in data visualization,* in *Expanding the Frontiers of Visual Analytics and Visualization* (Springer London, London, 2012) Chap. 6, pp. 81–109.

[139] R. van Pelt and A. Vilanova, *Understanding blood-flow dynamics: Challenges in visualization,* IEEE Computer **46**, 60 (2013).

[140] T. McLoughlin, M. Edmunds, C. Tong, R. S. Laramee, I. Masters, G. Chen, N. Max, H. Yeh, and E. Zhang, *Visualization of input parameters for stream and pathline seeding,* International Journal of Advanced Computer Science and Applications **6** (2015), 10.14569/IJACSA.2015.060417.

[141] J. Krüger, P. Kipfer, P. Konclratieva, and R. Westermann, *A particle system for interactive visualization of 3d flows,* IEEE Transactions on Visualization and Computer Graphics **11**, 744 (2005).

[142] W. Engelke, K. Lawonn, B. Preim, and I. Hotz, *Autonomous particles for interactive flow visualization,* Computer Graphics Forum **38**, 248 (2018), https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.13528 .

[143] R. Ihaka, *Colour for presentation graphics,* in *DSC 2003: Proceedings of the 3rd International Workshop on Distributed Statistical Computing* (2003).

[144] C. Brewer, *Color use guidelines for data representation,* in *Proceedings of the Section on Statistical Graphics* (American Statistical Association, 1999) pp. 55–60.

[145] M. D'Zmura, P. Colantoni, and J. Hagedorn, *Perception of color change,* Color Research and Application **26**, S186 (2001).

[146] M. Kersten-Oertel, S. Chen, and D. Collins, *An evaluation of depth enhancing perceptual cues for vascular volume visualization in neurosurgery,* IEEE Transactions on Visualization and Computer Graphics **20**, 391 (2014).

[147] K. Graf, M. Suter, J. Hagger, E. Meier, P. Meuret, and D. Nüesch, *Perspective terrain visualization – a fusion of remote sensing, GIS, and computer graphics,* Computers and Graphics **18**, 795 (1994).

[148] T. Ropinski, F. Steinicke, and K. Hinrichs, *Visually supporting depth perception in angiography imaging,* in *Smart Graphics: 6th International Symposium* (2006) pp. 93–104.

[149] T. Luft, C. Colditz, and O. Deussen, *Image enhancement by unsharp masking the depth buffer,* ACM Transactions on Graphics **25**, 1206 (2006).

[150] M. Everts, H. Bekker, J. Roerdink, and T. Isenberg, *Interactive illustrative line styles and line style transfer functions for flow visualization,* CoRR **abs/1503.05787** (2015).

[151] D. Fowler and C. Ware, *Strokes for representing univariate vector field maps,* in *Proceedings of Graphics Interface '89*, GI '89 (1989) pp. 249–253.

[152] T. Behrens, M. Woolrich, M. Jenkinson, H. Johansen-Berg, R. Nunes, S. Clare, P. Matthews, J. Brady, and S. Smith, *Characterization and propagation of uncertainty in diffusion-weighted MR imaging,* Magnetic Resonance in Medicine **50**, 1077 (2003).

[153] S. M. Smith, M. Jenkinson, M. W. Woolrich, C. F. Beckmann, T. E. Behrens, H. Johansen-Berg, P. R. Bannister, M. D. Luca, I. Drobnjak, D. E. Flitney, R. K. Niazy, J. Saunders, J. Vickers, Y. Zhang, N. D. Stefano, J. M. Brady, and P. M. Matthews, *Advances in functional and structural mr image analysis and implementation as FSL,* NeuroImage **23**, S208 (2004), mathematics in Brain Imaging.

[154] E. S. Farag, P. van Ooij, N. H. L. C. de Hoon, S. M. Boekholdt, E. L. Schade, R. N. Planken, A. Vilanova, M. G. Hazekamp, A. J. Nederveen, B. A. J. M. de Mol, D. R. Koolbergen, and J. Kluin, *Remodeling versus reimplantation in valve sparing aortic root replacement: insights from a 4-dimensional flow MRI study,* Unpublished, submitted for publication in the European Journal of Cardio-Thoracic Surgery (2018).

[155] M. A. Sarsam and M. Yacoub, *Remodeling of the aortic valve anulus.* The Journal of thoracic and cardiovascular surgery **105**, 435 (1993).

[156] T. E. David and C. M. Feindel, *An aortic valve-sparing operation for patients with aortic incompetence and aneurysm of the ascending aorta.* The Journal of thoracic and cardiovascular surgery **103**, 617 (1992).

[157] D. Tian, M. Rahnavardi, and T. D. Yan, *Aortic valve sparing operations in aortic root aneurysms: remodeling or reimplantation?* Annals of cardiothoracic surgery **2**, 44 (2013).

[158] T. H. Oechtering, C. F. Hons, M. Sieren, P. Hunold, A. Hennemuth, M. Huellebrand, J. Drexl, M. Scharfschwerdt, D. Richardt, H.-H. Sievers, J. Barkhausen, and A. Frydrychowicz, *Time-resolved 3-dimensional magnetic resonance phase contrast imaging (4d flow mri) analysis of hemodynamics in valve-sparing aortic root repair with an anatomically shaped sinus prosthesis.* The Journal of thoracic and cardiovascular surgery **152**, 418 (2016).

[159] J. W. van Rijswijk, A. Vink, J. R. Martina, F. Z. Ramjankhan, R. Goldschmeding, N. de Jonge, and J. Kluin, *Pathology of aortic valve remodeling after continuous-flow left ventricular assist device support.* (2017).

[160] Y. Sahasakul, W. D. Edwards, J. M. Naessens, and A. J. Tajik, *Age-related changes in aortic and mitral valve thickness: implications for two-dimensional echocardiography based on an autopsy study of 200 normal human hearts.* The American journal of cardiology **62**, 424 (1988).

[161] A. J. M. Broos, N. H. L. C. d. Hoon, P. J. H. d. Koning, R. J. v. d. Geest, A. Vilanova, and A. C. Jalba, *A framework for fast initial exploration of PC-MRI cardiac flow,* in *Eurographics Workshop on Visual Computing for Biology and Medicine*, edited by S. Bruckner, B. Preim, A. Vilanova, H. Hauser, A. Hennemuth, and A. Lundervold (The Eurographics Association, 2016).

[162] M. Elbaz, R. van der Geest, E. Calkoen, de A. Roos, B. Lelieveldt, A. Roest, and J. Westenberg, *Assessment of viscous energy loss and the association*

with three-dimensional vortex ring formation in left ventricular inflow: In vivo evaluation using four-dimensional flow MRI, Magnetic Resonance in Medicine (2016), 10.1002/mrm.26129.

[163] A. Gooch, B. Gooch, P. Shirley, and E. Cohen, *A non-photorealistic lighting model for automatic technical illustration,* in *Proceedings of the 25th annual conference on Computer graphics and interactive techniques* (ACM, 1998) pp. 447–452.

[164] E. Lum and K. Ma, *Hardware-accelerated parallel non-photorealistic volume rendering,* in *Proceedings of the 2nd international symposium on Non-photorealistic animation and rendering* (ACM, 2002).

[165] E. Calkoen, P. de Koning, N. Blom, L. Kroft, A. de Roos, R. Wolterbeek, A. Roest, and J. Westenberg, *Disturbed intracardiac flow organization after atrioventricular septal defect correction as assessed with 4D flow magnetic resonance imaging and quantitative particle tracing,* Investigative radiology **50**, 850 (2015).

[166] A. Fuhrmann and E. Gröller, *Real-time techniques for 3D flow visualization,* in *Proceedings of the conference on Visualization'98* (1998) pp. 305–312.

[167] L. Li and H.-W. Shen, *Image-based streamline generation and rendering,* IEEE Transactions on Visualization and Computer Graphics , 630 (2007).

[168] N. H. L. C. de Hoon, R. van Pelt, A. Jalba, and A. Vilanova, *4D MRI flow coupled to physics-based fluid simulation for blood-flow visualization,* Computer Graphics Forum **33**, 121 (2014).

[169] N. H. L. C. de Hoon, A. C. Jalba, E. Eisemann, and A. Vilanova, *Temporal interpolation of 4D PC-MRI blood-flow measurements using bidirectional physics-based fluid simulation,* in *Eurographics Workshop on Visual Computing for Biology and Medicine*, edited by S. Bruckner, B. Preim, A. Vilanova, H. Hauser, A. Hennemuth, and A. Lundervold (The Eurographics Association, 2016).

[170] M. Ghil and P. Malanotte-Rizzoli, *Data assimilation in meteorology and oceanography,* Advances in Geophysics **33**, 141 (1991).

[171] H. Zhao, *A fast sweeping method for Eikonal equations,* Mathematics of Computation **74**, 603 (2005).

[172] R. A. Gingold and J. J. Monaghan, *Smoothed particle hydrodynamics: theory and application to non-spherical stars,* Monthly Notices of the Royal Astronomical Society **181**, 375 (1977).

[173] M. Toma, C. H. Bloodworth, D. R. Einstein, E. L. Pierce, R. P. Cochran, A. P. Yoganathan, and K. S. Kunzelman, *High-resolution subject-specific mitral valve imaging and modeling: experimental and computational methods,* Biomechanics and Modeling in Mechanobiology **15**, 1619 (2016).

[174] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen, *LAPACK Users' Guide*, 3rd ed. (Society for Industrial and Applied Mathematics, 1999).

[175] C. Horvath and W. Geiger, *Directable, high-resolution simulation of fire on the GPU,* ACM Transactions on Graphics **28**, 41:1 (2009).

[176] C. A. Taylor, T. Hughes, and C. Zarins, *Finite element modeling of blood flow in arteries,* Computer Methods in Applied Mechanical Engineering **7825**, 155 (1998).

[177] R. Bammer, T. Hope, M. Aksoy, and M. Alley, *Time-resolved 3D quantitative flow MRI of the major intracranial vessels: Initial experience and comparative evaluation at 1.5T and 3.0T in combination with parallel imaging.* Magnetic Resonance in Medicine **57**, 127 (2007).

[178] E. J. Potchen and E. M. Haacke, *Magnetic resonance angiography: concepts & applications* (Mosby-Year Book, 1993).

[179] V. Verma and A. Pang, *Comparative flow visualization,* IEEE Transactions on Visualization and Computer Graphics **10**, 609 (2004).

[180] M. Duponcheel, P. Orlandi, and G. Winckelmans, *Time-reversibility of the Euler equations as a benchmark for energy conserving schemes,* Journal of Computational Physics **227**, 8736 (2008).

[181] L. Antiga, M. Piccinelli, L. Botti, B. Ene-Iordache, A. Remuzzi, and D. Steinman, *An image-based modeling framework for patient-specific computational hemodynamics,* Medical & Biological Engineering & Computing **46**, 1097 (2008).

[182] N. de Hoon, A. Jalba, E. Farag, P. van Ooij, A. Nederveen, E. Eisemann, and A. Vilanova, *Data assimilation for full 4d phase contrast magnetic resonance imaging measurements: Physics-based denoising and interpolation,* Computer Graphics Forum (2020), http://dx.doi.org/10.1111/cgf.14088.

[183] J. Nocedal, *Updating quasi-newton matrices with limited storage,* Mathematics of Computation **35**, 773 (1980).

[184] D. C. Liu and J. Nocedal, *On the limited memory BFGS method for large scale optimization,* Mathematical Programming **45**, 503 (1989).

[185] R. J. Hogan, *Fast reverse-mode automatic differentiation using expression templates in C++,* ACM Trans. Math. Softw. **40**, 26:1 (2014).

[186] G. Z. Yang, P. Burger, P. J. Kilner, S. P. Karwatowski, and D. N. Firmin, *Dynamic range extension of cine velocity measurements using motion registered spatiotemporal phase unwrapping,* Journal of Magnetic Resonance Imaging **6**, 495 (1996).

[187] R. Giering and T. Kaminski, *Recipes for adjoint code construction,* ACM Trans. Math. Softw. **24**, 437 (1998).

[188] P. J. van Ooij, A. Guédon, C. Poelma, J. J. Schneiders, M. C. M. Rutten, H. A. Marquering, C. B. L. M. Majoie, E. Vanbavel, and A. J. Nederveen, *Complex flow patterns in a real-size intracranial aneurysm phantom: phase contrast MRI compared with particle image velocimetry and computational fluid dynamics.* NMR in biomedicine **25**, 14 (2012).

[189] P. van Ooij, J. J. M. Zwanenburg, F. Visser, C. B. Majoie, E. van Bavel, J. Hendrikse, and A. J. Nederveen, *Quantification and visualization of flow in the Circle of Willis: Time-resolved three-dimensional phase contrast MRI at 7T compared with 3T,* Magnetic Resonance in Medicine **69**, 868 (2013).

Curriculum Vitae

Niels Hendrikus Louis Cornelis de Hoon

11-01-1989 Born in Etten-Leur, The Netherlands.

Education

2000–2007 Katholieke Scholengemeenschap Etten-Leur (KSE), Etten-Leur, The Netherlands

2007–2012 Bachelor Technische Informatica
Eindhoven University of Technology (TU/e)

2011–2013 Master Computer Science and Engineering (CSE)
Eindhoven University of Technology (TU/e)

2014–2019 PhD. Computer Science
Delft University of Technology (TU Delft)
Thesis: PC-MRI Blood-Flow Measurements - Visualization and Data Assimilation
Promotor: Dr. A. Vilanova
Promotor: Prof. dr. E. Eisemann
Co-promotor: Dr. A.C. Jalba

Work

2013 SeaChange
Software Engineer – ERP en CRM development
Eindhoven, The Netherlands

2013–2014 Profit
Consultant at ASML – Metrology
Eindhoven, The Netherlands

List of Publications

8. *Remodeling versus reimplantation technique in valve sparing aortic root replacement* (in submission)

7. **N.H.L.C. de Hoon**, A.C. Jalba, E.S. Farag, P. van Ooij, A.J. Nederveen, E. Eisemann and A. Vilanova *Data Assimilation for full 4D Phase Contrast Magnetic Resonance Imaging measurements: Physics-based denoising and interpolation*, Computer Graphics Forum (CGF), (2020).

6. **N.H.L.C. de Hoon**, K. Lawonn, A.C. Jalba, E. Eisemann and A. Vilanova, *InkVis: A High-Particle-Count Approach for Visualization of Phase-Contrast Magnetic Resonance Imaging Data*, Eurographics Workshop on Visual Computing for Biology and Medicine (VCBM), (2019).

5. J. Vendrik, E.S. Farag, **N.H.L.C. de Hoon**, J. Kluin and J. Baan Jr., *Presence of aortic root vortex formation after TAVI with CENTERA confirmed using 4D-flow magnetic resonance imaging*, The International Journal of Cardiovascular Imaging **34**, 1947 (2018).

4. **N.H.L.C. de Hoon**, E. Eisemann and A. Vilanova, *From a User Study to a Valid Claim: How to Test Your Hypothesis and Avoid Common Pitfalls*, EuroVis Workshop on Reproducibility, Verification, and Validation in Visualization (EuroRV3), (2017).

3. A.J.M. Broos, **N.H.L.C. de Hoon**, P.J.H. de Koning, R.J. van der Geest, A. Vilanova and A.C. Jalba, *A Framework for Fast Initial Exploration of PC-MRI Cardiac Flow*, Eurographics Workshop on Visual Computing for Biology and Medicine (VCBM), (2016).

2. **N.H.L.C. de Hoon**, A.C. Jalba, E. Eisemann and A. Vilanova, *Temporal Interpolation of 4D PC-MRI Blood-flow Measurements Using Bidirectional Physics-based Fluid Simulation*, Eurographics Workshop on Visual Computing for Biology and Medicine (VCBM), (2016).

1. **N.H.L.C. de Hoon**, R. van Pelt, A.C. Jalba and A. Vilanova, *4D MRI flow coupled to physics-based fluid simulation for blood-flow visualization*, Computer Graphics Forum **33**, 3 (2014).